October 7-10 • Ceará • Brazil

34th Brazilian Symposium on DATABASES

SBBD|2019

PROCEEDINGS COMPANION

October 7-10 • Ceará • Brazil

34th Brazilian Symposium on DATABASES

PROCEEDINGS COMPANION

Promotion
Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization
Departamento de Computação, Universidade Federal de Ceará– UFC

Steering Committee Chair
Bernadete Farias Lóscio (UFPE, Brazil)

Local Chair
José Maria da Silva Monteiro Filho (UFC, Brazil)

Program Committee Chairs
Full Paper: Carina F. Dorneles (UFSC, Brazil)
Short Paper: Fábio Porto (LNCC, Brazil)
Demos and Applications Chair: Robson L. F. Cordeiro (ICMC-USP, Brazil)
Thesis and Dissertation Workshop Chair: Jonice Oliveira (UFRJ, Brazil)
Tutorials Chair: Altigran Soares da Silva (UFAM, Brazil)
Short course Chair: Maria Cláudia Cavalcanti (IME, Brazil)
Workshop Chair: José Antônio Macedo (UFC, Brazil)
Thesis and Dissertation Contest Chair: Caetano Traina Jr. (USP, Brazil)
Graduation Student Workshop Chair: Ticiana Linhares (UFC, Brazil)

B839
Brazilian Symposium on Databases (SBBD 2019) (25.: 2019 october 07-10, 2019 –Fortaleza, CE)
Proceedings of 34nd Brazilan Symposium on Databases - SBBD 2019 [recurso eletrônico] / Organização: José Maria da Silva Monteiro Filho, Robson Leonardo Ferreira Cordeiro, Jonice de Oliveira Sampaio, Ticiana Linhares Coelho da Silva, Caetano Traina Junior - Fortaleza: SBC, 2019.

447p. v.2

Modo de acesso: http://sbbd.org.br/2019/

ISSN: 2016-5170

1. Computação - Congressos. 2. Bases de Dados– Congressos. I. José Maria da Silva Monteiro Filho. II. Sociedade Brasileira de Computação. III. Título.

CDD: 005

Elaborada pela Biblioteca da Universidade Federal do Ceará Campus Crateús (BCCR).

Message from the Local Organization Committee Chairs

Welcome to the 34th Brazilian Symposium on Databases and to Fortaleza, Ceará! The Brazilian Symposium on Databases is the official database event of the Brazilian Computer Society (SBC) and the largest venue in Latin America for presentation and discussion of research results in the database domain. The 34th edition of the symposium (SBBD 2019) was held in Fortaleza, in the state of Ceará, from October 7th to 10th, 2019. The local organization was performed by the Federal University of Ceará (UFC) through the Computer Science Department (DC). This year, for the first time, SBBD had the Symposium on Knowledge Discovery, Mining and Learning (KDMiLe); the Brazilian Symposium on Bioinformatics (BSB) and the ACM Latin American School on Recommender Systems (LARS) as co-located events providing a rich environment for the discussion of researches of their interrelated areas.

The SBBD 2019 program offers a wide variety of activities, suited for an audience ranging from undergraduate to Ph.D. students, database professionals, practitioners and researchers. The program includes: 3 invited talks and 2 tutorials, presented by distinguished speakers from Brazil, Chile and Germany; 9 technical sessions; 4 short courses about hot topics in the area, presented by specialists in their research fields; demos and applications session; posters sessions; industrial session, thesis and dissertations workshop; the biannual thesis and dissertations contest; 2 co-located workshops; the 3rd KDDBR (Brazilian Knowledge Discovery in Databases) competition; and a panel.

The excellence of SBBD 2019 program is the result of the competence and effort of a large community, which we gratefully acknowledge. The various sections of these proceedings list in detail those that contributed to the SBBD 2019 edition. We thank the symposium chairs and our colleagues of the local organization committee who donated their precious time to made SBBD 2019 a reality. We also thank the Computer Science Department (DC) of the Federal University of Ceará (UFC) and its Post-graduation Program (MDCC), which allowed their staff and students to help on the many tasks of the event preparation. We are also grateful to the SBC board for their support and to the steering committee members for their help, advice and support. Further, we thank the program committee members and external reviewers for the high-quality reviews, and the authors who submitted their papers to SBBD 2019. Finally, we are grateful to our sponsors. Without their support we would not be able to organize this annual event that brings together our community. We hope you all enjoy SBBD 2019 in Fortaleza, Ceará!

José Maria da Silva Monteiro Filho, UFC
SBBD 2019 Local Organization Committee Chair

Table of Contents

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Demos and Applications Track

PROCEEDINGS

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

Program Chair

Robson L. F. Cordeiro (ICMC-USP, Brazil)

Editorial

The Brazilian Symposium on Databases (SBBD) is the largest venue in Latin America for presenting research results in the database domain. In its 34th edition, SBBD will be held in Fortaleza, Ceará, from October 7th to 10th, 2019.

The Demonstrations and Applications Session is organized since 2004 within SBBD. It has become an important venue for sharing prototype data management systems with the SBBD community. The session aims at revealing new approaches and systems that contribute to data management research among researchers, developers and professionals, from both academia and industry.

In this edition issue, we had 8 interesting demo papers selected from a total of 12. Each paper was evaluated by at least 3 reviewers selected from a committee of 36 researchers from both academia and industry.

The Demonstration and Application Session is the result of the collective effort of the SBBD community, which we gratefully acknowledge. First, we are very thankful to all authors of submitted papers for their interest in Demonstration and Application Session. Second, we would like to thank the reviewers for their high-quality evaluations.

Finally, we would like to thank the SBBD 2019 organizers for all local arrangements that provide the necessary infrastructure for Demonstration and Application Session. We hope you all enjoy SBBD Demonstration and Application in Fortaleza!

Robson L. F. Cordeiro (ICMC/USP)
Program Chair – SBBD 2019 – Demos and Applications

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Promotion

Brazilian Computer Society – SBC
SBD Database Steering Committee

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

SBBD Steering Commitee

Ângelo Brayner (UFC)
Bernadette Lóscio (UFPE) Steering Committee Chair
Carina Dorneles (UFSC)
Sérgio Lifschitz (PUC-Rio)
Fábio Porto (LNCC)
Carmem Hara (UFPR)

SBBD 2019 Commitee

Steering Committee Chair
Bernadette Lóscio (UFPE)

Local Chair:
José Maria da Silva Monteiro Filho (UFC, Brazil)

Full Paper Chair
Carina F. Dorneles (UFSC, Brazil)

Short Paper Chair
Fábio Porto (LNCC, Brazil)

Demos and Applications Chair:
Robson L. F. Cordeiro (ICMC-USP, Brazil)

Thesis and Dissertation Workshop Chair
Jonice Oliveira (UFRJ, Brazil)

Tutorials Chair
Altigran Soares da Silva (UFAM, Brazil)

Short course Chair
Maria Cláudia Cavalcanti (IME, Brazil)

Workshop Chair
José Antônio Macedo (UFC, Brazil)

Thesis and Dissertation Contest Chair
Caetano Traina Jr. (USP, Brazil)

Graduation Student Workshop Chair
Ticiana Linhares (UFC, Brazil)

Local Organization Committee
SBBD Local Chair: José Maria da Silva Monteiro Filho (DC/UFC)
Leonardo Oliveira Moreira (Instituto UFC Virtual/UFC)
Marum Simão Filho (UNI7)
Angelo Roncalli de Alencar Brayner (DC/UFC)
Javam de Castro Machado (DC/UFC)

Demos and Applications Chair Program Committee:
Agma Traina (USP)
Anderson Chaves Carniel (UFPR)
Anderson Ferreira (UFOP)
Angelo Brayner (UFC)
Caetano Traina Júnior (USP)
Carlos Eduardo Santos Pires (UFCG)
Cristina Ciferri (USP)
Damires Souza (IFPB)
Daniel de Oliveira (UFF)
Daniel Kaster (UEL)
Eduardo Bezerra (CEFET/RJ)
Eduardo Ogasawara (CEFET/RJ)
Elaine Sousa (USP)
Fabio Porto (LNCC)
Flávio R. C. Sousa (UFC)
Geraldo Xexéo (UFRJ)
Guilherme Alves (INRIA Nancy-Grand Est/Université de Lorraine)
Humberto Razente (UFU)
Jonas Dias (DELL EMC)
José Maria Monteiro (UFC)
José Antonio Macêdo(UFC)
José de Aguiar Moraes Filho (UNIFOR)
Jose Rodrigues Jr (USP)
Karin Becker (UFRJ)
Kary Ocaña (LNCC)
Leonardo Azevedo (IBM Research Brazil)
Luiz Henrique Zambom Santana (UFSC)
Marcela Ribeiro (UFSCar)
Marcio Oikawa (UFABC)

Marcos Bedo (UFF)
Maria Camila Nardini Barioni (UFU)
Maristela Holanda (UNB)
Mirella Moro (UFMG)
Renata Galante (UFRGS)
Ricardo Torres (UNICAMP)
Robson Leonardo Ferreira Cordeiro (USP)
Rodrigo Monteiro (UFF)
Ronaldo Correia (UNESP)
Ronaldo Mello (UFSC)
Victor de Almeida (UFF/Petrobras)

Additional Reviewers:
Anderson Chaves Carniel (Federal University of Technology - Paraná)
Guilherme Alves (INRIA Nancy-Grand Est, Université de Lorraine - France)
Luiz Henrique Zambom Santana (Universidade Federal de Santa Catarina -Brazil)

Table of Contents (Demos and Applications Track)

DSS: A Data Science Suite

Rafael S. Pereira[1], **Fabio Porto**[1]

[1]Laboratório Nacional de Computação Cientifica (LNCC), Data Extreme Lab (DEXL)
CEP: 22651-075 – Petrópolis – RJ – Brazil

{rpereira,fporto}@lncc.br

Abstract. *This paper presents a set of applications with graphical interface for data science tasks. In the course of the paper we shall be discussing applications for data visualization, machine and deep learning, natural language processing, Graph data analysis and time series analysis. This suite of applications runs in a service at LNCC provided by the DEXL Lab Group and can be individually installed in a local machine using docker images.*

1. Introduction

In current times where data is being generated in higher and higher volumes, analyzing these data has become a matter of business and science success. Professionals in different fields are challenged to use the tools provided by data science to leverage the amount of available information. However, this is a daunting task as the type of data that must be analysed is different for each field coupled with the fact that not all professionals have the coding skills necessary to express the analysis required for their specific problem. The challenge increases as data science has many different facets, being able to work with tabular data, texts, audio, images, and many other types of data. Because of this, a non-expert user faces various types of tools that must be mastered, one for each type of data.

This paper presents a suite of applications for data science that lets the user understand and train models on different types of data without needing to know how to code. By means of the Data Science Suite (DSS) we expect to extend the applicability of data science techniques to a broader public bringing state-of-the-art algorithms to improve business and decision-making. Throughout this paper we shall be discussing how these services work and what they provide.

The remaining of this paper is structured as follows: Firstly we briefly contextualize *DSS* by introducing some similar initiatives, in the related work Section. Next, we present each individual service provided by the suite, illustrated with examples of their use. Finally, we make some considerations and point to the URL where the service can be reached.

2. Related Work

Overall, the services provided by DSS can be found in different tools. DIVE[Hu et al.], for instance, from MIT features data visualization, RapidMiner[Cha] provides natural language processing tools and Gephi[Bastian et al. 2009] provides Graph Analysis tools for visualization and calculating properties on Graph data. Cloud providers like GCP, Amazon and Azure provide automated machine learning and Deep Learning with no coding required as well.

3. DSS Analysis Services

Currently, *DSS* counts with the following services:

- Interactive Data Visualization and Exploration.
- Comparison of PDF files for similarity
- Finding themes contained in text
- Sentiment Analysis in text
- Graph data Analysis
- Time Series Analysis
- Graphical interface for Supervised and Unsupervised Machine Learning.
- Deep learning models evaluation on unseen data
- Classification application with pre trained models
- Object detection models in a web interface.

3.1. Data visualization and Exploration

This service aims to provide users with the ability to visualize and understand their own tabular data like the service provided by DIVE.
Data visualization is a ubiquitous tool necessary for all different kinds of industry. It is considered that only with the insights obtained by looking at the data can informed decisions be made on any process.

This Web application requests a tabular file and lets users explore the dataset with interactive data visualizations. Examples of analysis include: observe statistical information about its attributes; evaluate missing data; check which equations best fit a curve the user is visualizing as well as observing correlations among different attributes of the dataset .

In Figure 1 one can see the app with a dataset of kickstarter data loaded into it.

Figura 1. Visualization and Exploration Service

Figura 2. TextAnalysis App

3.2. Text Analytics

This service offers users a tool to understand the content of a PDF file applying the natural language processing algorithms developed for data science.

Natural language processing becomes an important technique for many different industries, specially the marketing section of them, since we can analyze data from texts (eg . product reviews) written by potential costumers to understand how a given product has been received or could be received by different target audiences.

This Web Application lets the user provide a PDF file as an input, and see a distance matrix of the words that appear with a minimum frequency, using the mean metric the user can see which words appear together most of the time, while the standard deviation shows the user which words always appear equidistant to one another. The clustering of this matrix can show the user the different topics the file talks about, and two different files can be compared, both on the clustering part and their distance matrix to check for similarities between these files. Figure 2 shows us this service running.

3.3. Sentiment Analysis

This application is also focused on the natural language processing part of data science, it lets the user explore the sentiment contents of a file to understand its content trough interactive graphs, both it and Text Analytics are related to the services provided by RapidMiner.

This Web application requests a PDF file and lets the user see the local sentiment of every n words , evaluate the cumulative sentiment which is the cumulative sum of these local sentiments to determine the overall sentiment of the file as well as its curve. This can be done both by looking at positive - negative sentiment as well as looking at each of the eight sentiments the program analyzes. Such analysis enables the user to observe whether the content of the file is positive or negative, as well as highlighting its principal sentiments.

Figura 3. Sentiment Analysis App

3.4. Machine Learning

This Application is focused on the [techopedia] AutoML part of Data science. It is focused on letting the user train models for regression and classification on their own data and downloading these afterwards. These kinds of services are mostly provided by the main cloud providers by a price nowadays.

This Web Application lets the user input a tabular file and train supervised and unsupervised machine learning models on this data via a graphical interface, after training the user can see the result's trough metrics like accuracy for classification or RMSE for regression algorithms, and the trained model can be downloaded trough in a .Rdata format which can be loaded into R and used with caret for production.

Figure 4 shows this application working

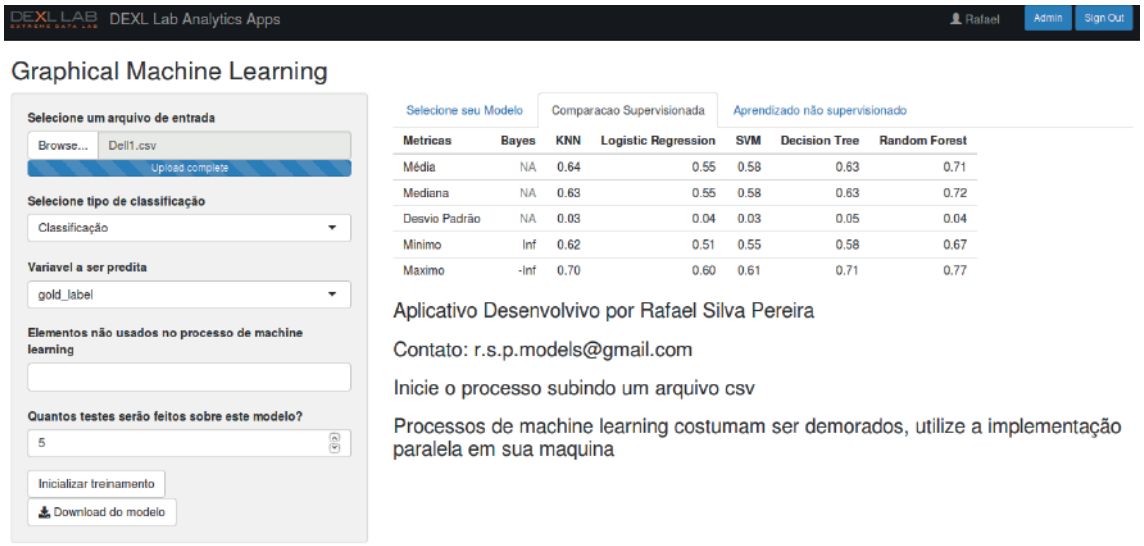


Figura 4. Machine Learning App

3.5. Time Series

This application is focused on time series data, it lets the user analyze and understand how a time series behaves based on many different metrics and visualize this behavior, many different fields are dependent on time series analytics like the healthcare and finance industry.

The time series Web Application expects a file with only one column representing a time series with equidistant points in the x axis , then the app lets the user visualize the time series trough many different metrics, Figure 5 shows this Application working.

Figura 5. Time Series App

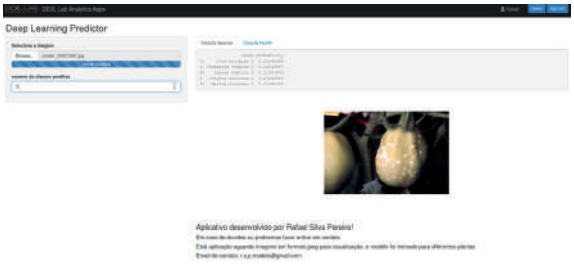


Figura 6. Deep Learning App

3.6. Deep Learning

Both Deep Learning and object detection services are provided to show how can we integrate trained models for production to the end user. These services are currently provided

by cloud platforms, as cited with respect to the *Machine Learning* application. Similarly, to the latter, this kind of service is useful in many different types of industries.

In Deep learning we integrate two different Web Apps using the tensorflow backend . The first one expects a *keras API* trained model as well as the respective *pickle* file format and then uses this model to predict images that are uploaded to it. The second one already hosts two trained models: one for Plant Classification into different species; the other for classifying plants between Healthy and Sick. Figure 6 shows us the results of a plant classification task.

3.7. Graph Analysis

In many different fields, the topology of a problem may be expressed by a graph, and when doing so many properties can be derived from this graph based on Graph Theory. This application automates the process of graph analysis by expecting the adjacency matrix of the graph, and letting the user see the value of many different properties of both the graph, the nodes, and the pair combinations of nodes.

Figura 7. Graph Analysis App

3.8. Object Detection

The object detection application uses a *YOLO* real-time object detection system implementation using the pretrained *darknet* model to run an object detection algorithm. The user must give to the application an image and the application will return this image with all objects detected marked with bounding boxes, Figure 8 shows the application running.

Figura 8. Object Detection App

4. Contributions and Conclusion

In this paper we presented several web applications that are running in the link https://dexl-analytics-apps.lncc.br/

Each of these applications lets the user perform a task of data science with different types of data without needing no knowledge of programming, this way the analysis may be done by many more people and way faster.

All these applications are provided without restrictions on dockerhub as well so if the user needs to analyze larger amounts of data that can not be done via the Data Extreme Lab service, the docker image may be downloaded locally and the only limitation is the end user hardware.

The technologies used to create this project were R[R Core Team 2014], Python[van Rossum 1995], Docker[Merkel 2014] and Shinyproxy

5. Future Work

DSS is in it first version, we intend to extend the initial prototype in a few dimensions. The following is a list of directions we intend to investigate. We shall explore the integration of a backend engine to support data scalability in face of large datasets. Another challenge is given the different applications to come-up with an strategy to help users define the parameter setting values, A version that supports storing analysis outputs to a database can also be looked at.

6. Acknowledgements

The authors would like to thank the support of Conselho Nacional de Desenvolvimento Científico e Tecnológico(CNPQ),This study was financed in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico(CAPES) -Finance Code 001.

Referências

Data mining use cases and business analytics applications. Chapman Hall/CRC.

Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks.

Hu, K., Orghian, D., and Hidalgo, C. Dive: A mixed-initiative system supporting integrated data exploration workflows. In *ACM SIGMOD Workshop on Human-In-the-Loop Data Analytics (HILDA)*.

Merkel, D. (2014). Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239).

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

techopedia. Automl.

van Rossum, G. (1995). Python tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

PAbS: Um Processador de Consultas SPARQL sobre Bases Distribuídas

Raqueline R. M. Penteado[1], Hugo Paulino B. Takiuchi[2], Carmem S. Hara[2]

[1] Departamento de Informática – Universidade Estadual de Maringá
Avenida Colombo 5.790 – 87.020-900 – Maringá, PR – Brasil

[2]Departamento de Informática – Universidade Federal do Paraná
Caixa Postal 19.081 – 81.531-990 – Curitiba, PR – Brasil
raque@din.uem.br, {hpbtakiuchi,carmem}@inf.ufpr.br

Abstract. *PAbS is an RDF-based system that supports both distributed storage and parallel and distributed processing of SPARQL queries. Query execution applies a hybrid communication method which combines two communication strategies: send-result and get-frag. This method aims at minimizing the communication cost of intermediary results among storage servers. In this demonstration, we present the query processing functionality of the system, along with the communication strategy chosen by PAbS whenever it requires data to be exchanged between servers.*

Resumo. *PAbS é um sistema que dá suporte ao armazenamento distribuído de bases RDF e faz o processamento paralelo e distribuído de consultas SPARQL usando um método de comunicação híbrido que combina duas estratégias: send-result e get-frag. O principal objetivo deste método é minimizar o custo de comunicação no processamento distribuído de consultas. Nesta demonstração são apresentadas as funcionalidades do sistema, mostrando a execução de consultas e a escolha de estratégias de comunicação pelo PAbS.*

1. Introdução

A *Web semântica* é uma extensão da *Web* que associa significado aos recursos disponíveis na rede, de forma que as máquinas sejam capazes de entender as informações e executar tarefas sofisticadas para os usuários. O consórcio *W3C*[1] definiu o RDF (*Resource Description Framework*) e SPARQL (*SPARQL Protocol and RDF Query Language*) como o modelo de dados e a linguagem de pesquisa padrão para a *Web semântica*, respectivamente. Uma base RDF é composta por um conjunto de triplas $(sujeito, predicado, objeto)$. Como o objeto de uma tripla pode desempenhar o papel de sujeito de outra, uma base RDF pode ser vista como um grafo, no qual sujeitos e objetos são representados por nodos ligados por arestas que representam os predicados, conforme mostra a Figura 1(b). Bases de dados RDF comerciais têm alcançado o tamanho de 1 trilhão de triplas[2]. Assim, o processamento eficiente de consultas SPARQL neste tipo de base é um grande desafio para os sistemas gerenciadores de dados RDF.

O armazenamento e processamento centralizado pode penalizar a escalabilidade em aplicações com grande volume de dados. Logo, sistemas com armazenamento distribuído têm sido propostos neste contexto. Nestes sistemas, tanto os dados quanto as

[1] http://www.w3.org

[2] http://www.w3.org/wiki / LargeTripleStores

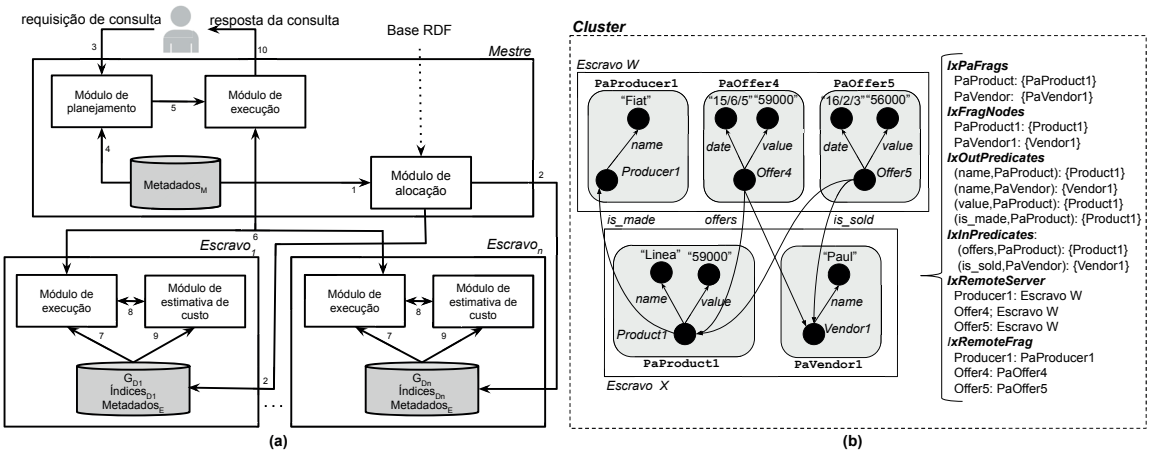


Figura 1. Arquitetura do sistema(a); Base RDF distribuída(b)

consultas são distribuídas entre servidores a fim de promover o processamento paralelo e distribuído. Porém, a troca de dados entre servidores durante o processamento distribuído afeta diretamente no seu custo [Ozsu and Valduriez 2011].

Este artigo demonstra o sistema *PAbS* (*Pattern Allocation-based System*), que se baseia nas técnicas apresentadas em [Penteado et al. 2016]. O sistema gera um plano de execução de consultas SPARQL usando o conhecimento prévio de como a base de dados RDF foi particionada. O particionamento implementa o conceito de fragmentos, que consistem em subgrafos da base. Fragmentos são utilizados como unidades de comunicação, quando uma consulta envolve dados armazenados em servidores distintos. Intuitivamente, na execução do plano, após a exploração de um fragmento de dados em um servidor, duas estratégias de comunicação podem ser consideradas para dar continuidade à execução: a requisição de fragmentos necessários de outros servidores (*get-frag*) ou o envio de resultados intermediários para outros servidores (*send-result*). O objetivo inovador do *PAbS* de dar suporte às duas estratégias de comunicação é minimizar o custo de comunicação entre servidores no processamento distribuído de consultas.

2. Trabalhos relacionados

A maioria dos sistemas RDF distribuídos adotam métodos de comunicação baseados na estratégia *send-result*. Sistemas que usam o *MapReduce*[3] no processamento distribuído de consultas baseiam-se nesta estratégia. Neste *framework*, um plano de execução é composto por tarefas de mapeamento e redução. Tarefas de redução recebem e processam resultados intermediários gerados por tarefas de mapeamento em servidores remotos. Já, em [Harbi et al. 2016] cada servidor executa um plano de consulta na sua totalidade, recuperando resultados intermediários por meio de requisições a servidores remotos.

Rya [Punnoose et al. 2012] e *MAPSIN* [Przyjaciel-Zablocki et al. 2012] são exemplos de sistemas que adotam métodos de comunição baseados na estratégia *get-frag*. Em *Rya*, consultas são executadas de maneira centralizada enquanto dados são recuperados de servidores remotos por meio de múltiplos *lookups* em um repositório distribuído Accumulo[4]. *MAPSIN* usa a mesma ideia no contexto do *MapReduce*.

[3]https://hadoop.apache.org/docs/r1.2.1/index.html
[4]https://accumulo.apache.org/

Por fim, [Penteado et al. 2016] propõe o método de comunicação *2ways* que viabiliza a escolha entre as duas estratégias de comunicação em tempo de execução. O sistema *PAbS* implementa este método.

3. O Sistema *PAbS*

PAbS adota uma arquitetura *mestre-escravo shared-nothing*, conforme mostra a Figura 1(a). O sistema envolve os módulos de *alocação de dados* e de *processamento de consultas* (*planejamento*, *execução* e *estimativa de custo*). A arquitetura do sistema adota o paralelismo entre servidores, no qual cada escravo executa requisições em paralelo com outros escravos, sem a exigência de sincronismo entre os mesmos. *PAbS* foi implementado em Java usando o protocolo TCP/IP na comunicação entre escravos. Dados foram armazenados em memória por meio do repositório *Berkeley DB*[5]. O código fonte do sistema está disponível em https://github.com/RaquelinePenteado/PAbS.

3.1. Alocação

O *módulo de alocação* é responsável por fragmentar e distribuir uma base RDF entre os escravos de um *cluster*. Padrões de alocação (PAs) definem padrões estruturais usados para particionar uma base RDF. A Figura 2(a) apresenta o grafo de sumarização G_S da base G_D da Figura 1(b). G_S possui quatro PAs que estão representados por linhas tracejadas e nomeados de acordo com a classe que os compõem. Os PAs adotam o padrão estrutural do tipo estrela, garantindo que cada sujeito que representa um recurso seja armazenado com os seus respectivos objetos literais. Fragmentos são instâncias de PAs que seguem sua estrutura. Por exemplo, os fragmentos *PaOffer4* e *PaOffer5* da Figura 1(b) são instâncias do padrão de alocação *PAOffer* da Figura 2(a). Os fragmentos são as unidades de comunicação do sistema e garantem co-alocação no armazenamento. Ou seja, nodos em um grafo RDF G_D que pertencem ao mesmo fragmento são alocados em um mesmo servidor e qualquer transmissão de dados entre os servidores envolve no mínimo um fragmento. Vale destacar que arestas de G_S que conectam nodos de um mesmo PA garantem que os nodos correspondentes da base G_D são alocados em um mesmo servidor. Arestas que conectam nodos de PAs distintos não fornecem esta garantia.

Considerando os fluxos numerados da Figura 1(a), dada uma base RDF, o mestre faz a fragmentação de acordo com PAs pré-definidos (1) e distribui os fragmentos gerados entre os escravos do *cluster* (2). Quanto ao armazenamento, *PAbS* adota um modelo nativo de grafos, no qual uma base RDF é definida como um conjunto de nodos com suas listas de adjacência *in* (arestas que incidem no nodo) e *out* (arestas que partem do nodo). Por exemplo, na base da Figura 1(b), o conjunto de arestas incidentes em *Producer1* é *in*:{(is_made, {Product1})} e o conjunto de arestas que partem, *out*:{(name,*"Fiat"*)}.

Durante o inserção de dados nos escravos, índices são criados para dar suporte às funcionalidades do sistema. Cada escravo armazena localmente em seu repositório índices referentes à sua base local. A Figura 1(b) mostra os índices do Escravo X. O índice *IxPaFrags* associa cada padrão de alocação (PA) a todos os fragmentos deste tipo. O índice *IxFragNodes* associa cada fragmento aos recursos nele contidos. *IxOutPredicates* e *IxInPredicates* indexam recursos por meio de predicados e seus PAs. *IxOutPredicates* considera as arestas (predicados) que partem dos recursos e *IxInPredicates*, as arestas

[5] http://www.oracle.com/technetwork/database/database-technologies/berkeleydb

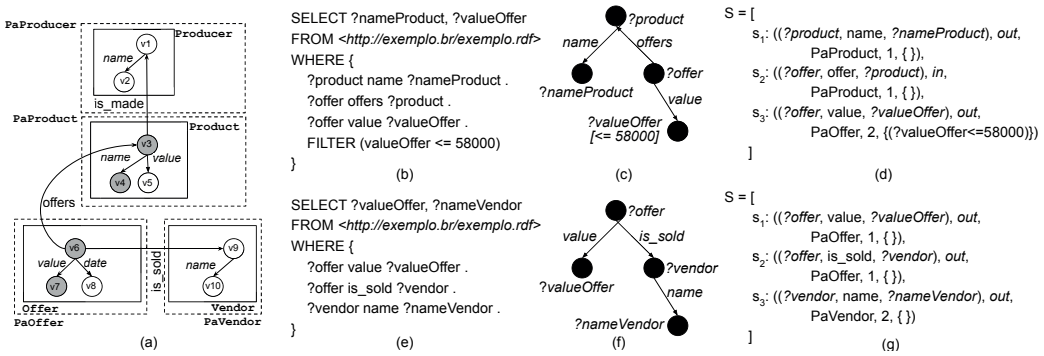


Figura 2. Grafo de sumarização(a); Consulta SPARQL Q_1(b); Grafo de Q_1(c); Plano de Q_1(d); Consulta SPARQL Q_2(e); Grafo de Q_2(f); Plano de Q_2(g)

que incidem nos recursos. Dada uma propriedade que conecta dois recursos alocados em escravos distintos, o índice *IxRemoteServer* permite que o um escravo obtenha o endereço físico do outro, possibilitando a exploração distribuída de grafos. Por fim, *IxRemoteFrag* permite a identificação de um fragmento remoto, dado o identificador de um recurso.

Além dos índices, cada escravo armazena localmente os metadados necessários para o seu funcionamento (Metadados$_E$). Metadados sobre o tamanho médio dos fragmentos de dados de G_D gerados pelo *módulo de alocação* são armazenados nos escravos. Estes dados são utilizados pelo *módulo de estimativa de custo* da abordagem de processamento de consultas. O mestre armazena em seu repositório o grafo de sumarização de G_D, dando suporte ao *módulo de alocação* e ao *módulo de planejamento* (Metadados$_M$).

3.2. Processamento de consultas

PAbS dá suporte a consultas SPARQL que contêm a cláusula SELECT, o operador AND (.), o operador FILTER e permite variáveis somente em sujeitos e objetos nos padrões de triplas. As consultas das Figuras 2(b) e 2(e) ilustram consultas deste tipo. A primeira consulta retorna, para cada produto, o seu nome e os valores de suas ofertas $\leq$ *58000*. A segunda retorna, para cada oferta, o seu valor e o nome de seu vendedor. Consultas SPARQL podem ser representadas por meio de grafos e o *PAbS* considera grafos de consultas conexos. O grafo G_Q da Figura 2(c) representa a consulta da Figura 2(b), no qual os sujeitos e os objetos dos padrões de triplas são representados por nodos e os predicados por arestas. O filtro da consulta está representado entre colchetes abaixo do nodo da variável apropriada. O grafo da Figura 2(f) representa a consulta da Figura 2(e). O processamento de consultas SPARQL pode ser transformado em um problema de casamento de grafos onde subgrafos de G_D que são homomórficos a G_Q são recuperados.

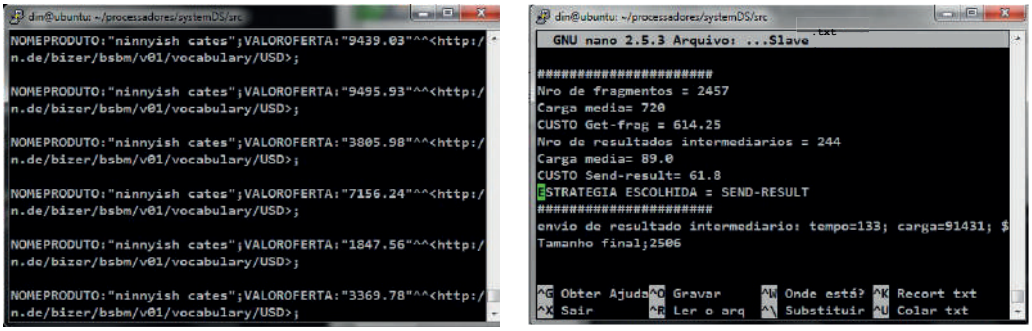
Voltando à arquitetura do sistema, dada uma requisição de consulta (3), o servidor mestre elabora o seu plano de execução usando metadados (4) e envia o plano da consulta para todos os escravos (5) e (6). A partir deste ponto, cada escravo inicia a execução do plano em paralelo usando o seu repositório local (7), trocando mensagens no *cluster* (6) por meio de uma estratégia de comunicação (8), que pode ser definida por meio de metadados (9) se necessário. Ao final, o servidor mestre recebe os resultados gerados pelos escravos (6), retornando o resultado final da consulta ao usuário (10). O processamento está dividido em duas partes, o planejamento (*módulo de planejamento*) e a execução de consultas (*módulo de execução*).

O planejamento baseia-se nas estruturas dos fragmentos definidos pelos PAs de G_S a fim de minimizar a comunicação entre servidores durante a execução de consultas. Como todos os nodos de um fragmento estão armazenados em um mesmo servidor, o plano de consulta gerado percorre as arestas internas do fragmento antes de passar para um novo fragmento. O subgrafo destacado na Figura 2(a) representa o subgrafo homomórfico ao grafo G_Q da Figura 2(c). As Figuras 2(d) e 2(g) mostram os planos gerados para as consultas das Figuras 2(b) e 2(e), respectivamente. Cada passo s_i do plano é uma tupla (a, dir, pat, id, f), onde: (**1**) a é um padrão de tripla de Q, (**2**) $dir \in \{in, out\}$; se $dir = out$ a exploração é do sujeito para o objeto. Caso contrário, a exploração é do objeto para o sujeito; (**3**) pat é o PA que contém a; (**4**) id é um identificador único associado ao PA. Todos os passos com um mesmo id compõem um bloco de exploração dentro de um mesmo fragmento. (**5**) f é um conjunto de filtros definidos em a. O resultado de uma consulta SPARQL consiste de um conjunto de mapeamento das variáveis da cláusula SELECT para nodos do grafo G_D. Assim, cada passo s_i de um plano gera um conjunto de mapeamentos para as variáveis no padrão de tripla $s_i.pat$ que satisfaz o filtro $s_i.f$.

Para o processamento, cada escravo do *cluster* inicia o plano de execução em paralelo. Considere o plano da Figura 2(d). O primeiro passo do plano é s_1*: ((?product, name, ?nameProduct), out, PaProduct, 1,* $\{\}$). Ele define que a exploração de uma tripla com predicado *name* em fragmentos de *PaProduct* deve ser realizada do sujeito para o objeto (direção *out*). A execução deste plano no escravo X da Figura 1(b) utiliza o índice *IxOut-Predicates* para obter o nodo *Product1* e associá-lo à variável *?product*. Ele passa a ser um ponto inicial de exploração. A partir daí, cada passo do plano adiciona novas associações de variáveis a nodos, caminhando no grafo RDF. Se o padrão de tripla não é encontrado ou se a nova associação não satisfaz os filtros, ele é removido do conjunto resultado. Continuando o exemplo, o escravo X processa o passo s_2, gerando o resultado parcial $\{(?product \mapsto Product1, ?nameProduct \mapsto \text{“}Linea\text{”}, ?offer \mapsto \{Offer4, Offer5\})\}$. O processamento de s_3 pode continuar ou não em X uma vez que este passo envolve um PA distinto do passo anterior. O índice *IxRemoteServer* auxilia na localização dos nodos das ofertas *Offer4* e *Offer5*. No exemplo, as ofertas estão armazenadas no servidor W. Logo, X se comunica com W continuando a execução da consulta. Neste momento, ou X requisita os fragmentos das ofertas para W e continua o processamento localmente (usando a estratégia *get-frag*), ou o mapeamento gerado é enviado para W continuar a execução do plano (usando a estratégia *send-result*). O índice *IxRemoteFrag* define a recuperação dos fragmentos remotos *PaOffer4* e *PaOffer5* para *get-frag*. A escolha da estratégia é feita por meio de ***funções de custo*** e depende do número de mensagens e do volume de dados a ser transmitido entre os escravos. O objetivo é minimizar o custo de comunicação entre os escravos. Por fim, os resultados parciais gerados para *Offer4* são descartados em s_3, uma vez que a oferta não satisfaz o filtro $s_3.f$. O resultado da consulta é $\{$*(nameProduct: “Linea”; valueOffer: 56000)*$\}$. Mais detalhes sobre o processamento, estratégias de comunicação e funções de custo podem ser encontrados em [Penteado et al. 2016].

4. Demonstração

Os planos das Figuras 2(d) e 2(g) foram executados em uma base RDF distribuída em um *cluster* com dois escravos. Fragmentos da base foram gerados por meio do grafo de sumarização da Figura 2(a) e alocados aleatoriamente no *cluster*. A Figura 3(a) mostra os resultados retornados pelo *PAbS* na execução do primeiro plano. A Figura 3(b) mostra



(a) (b)

Figura 3. Retorno do primeiro plano(a); *Log* da execução do primeiro plano(b)

a escolha da estratégia de comunicação feita por um escravo do *cluster*. A estratégia que obteve o menor valor na sua função de custo foi escolhida. Logo, o escravo usou *send-result* na transição entre os passo s_2 e s_3 do plano, uma vez que um produto relaciona-se com várias ofertas. O escravo enviou 244 resultados intermediários para o escravo remoto levando 133 ms. A escolha de *get-frag* envolveria a recuperação de 2.457 fragmentos do escravo remoto levando 2.819 ms. Para a coleta do tempo de *get-frag*, o uso da estratégia foi definida manualmente no *PAbS*.

Na execução do segundo plano, o escravo escolheu *get-frag* na transição entre s_2 e s_3, uma vez que várias ofertas relacionam-se com um vendedor. O escravo recuperou 13 fragmentos do escravo remoto em 140 ms. A escolha de *send-result* envolveria o envio de 2.274 resultados intermediários para o escravo remoto levando 399 ms. Novamente, o uso de *send-result* foi feito manualmente no sistema.

5. Conclusão

Este trabalho demonstra o sistema *PAbS*, que processa consultas SPARQL em uma base RDF distribuída. As funcionalidades do sistema baseiam-se no conceito de padrão de alocação. Enquanto a maioria dos sistemas RDF distribuídos adota uma única estratégia de comunicação, *PAbS* faz a escolha da estratégias de comunicação durante a execução de consultas. Trabalhos futuros envolvem a extensão de *PAbS* a fim de dar suporte para o processamento de consultas SPARQL em sistemas RDF federados.

Referências

Harbi, R., Abdelaziz, I., Kalnis, P., Mamoulis, N., Ebrahim, Y., and Sahli, M. (2016). Accelerating SPARQL Queries by Exploiting Hash-based Locality and Adaptive Partitioning. *The VLDB Journal*, 25(3):1–26.

Ozsu, M. T. and Valduriez, P. (2011). *Principles of Distributed Database Systems, 3rd Edition*.

Penteado, R. R. M., Schroeder, R., and Hara, C. S. (2016). Exploring controlled RDF distribution. In *IEEE CloudCom 2016, Luxembourg, December 12-15, 2016*, pages 160–167.

Przyjaciel-Zablocki, M., Schätzle, A., Hornung, T., Dorner, C., and Lausen, G. (2012). Cascading Map-side Joins over HBase for Scalable Join Processing. *CoRR*, abs/1206.6293.

Punnoose, R., Crainiceanu, A., and Rapp, D. (2012). Rya: A Scalable RDF Triple Store for the Clouds. In *Proceedings of the 1st Cloud-I*, pages 4:1–4:8, New York, NY, USA.

From Data Requirements to Health Applications: A Tool for Dynamic Generation of Data Schemas and Graphical User Interfaces Using Archetypes

André Magno C. de Araújo[1], Valéria C. Times[2], Marcus U. Silva[2]

[1]Departament of Information Systems
Federal University of Alagoas (UFAL) – Penedo, AL – Brazil

[2]Center for Informatics
Federal University of Pernambuco (UFPE) – Recife, PE – Brazil

andre.araujo@penedo.ufal.br,vct@cin.ufpe.br, mus@cin.ufpe.br

Abstract. *This paper presents a computational tool to dynamically create data schemas and graphical user interfaces from data attributes, terminologies and constraints specified in archetypes. Template4EHR allows end users to build health applications based on the Electronic Health Record specifications defined in archetypes, thus minimizing their dependency on a software development team. The main features of the tool include the creation of relational and NoSQL data schemas and the generation and customization of graphical user interfaces.*

1. Introduction

Over the past few years, the software industry, governmental and academic institutions have debated the use of health standards in developing solutions that improve service quality, increase the productivity of health professionals and popularize access to the Electronic Health Record (EHR) [Lin et.al 2016, Lee et.al 2013]. This initiative aims to improve the flexibility and extensibility of Health Information Systems (HIS) and to enable professionals to effectively contribute to the development of HIS.

As a common characteristic, a software system changes over time and must adapt to the new requirements of its application domain, even while it is running. An HIS deals with data requirements that continually change or are specialized after a short period of time. Consequently, HIS built through traditional software development methodologies are expensive to maintain [Jeffrey et.al 2013]. Generally, changes in a health application require effort and the dependence on a programming team. It must also be noted that HIS are not designed to allow dynamic changes, that is, they do not adapt to the context of the problem domain, nor do they allow end users to create new instances of an application for other domains or to develop new functionalities.

In this context, the openEHR dual architecture represents an important initiative aimed at improving the flexibility, maintainability and extensibility of HIS [Georg et.al 2013]. The specification of this architecture is produced through archetypes and templates. An archetype can be defined as a computational expression represented by domain constraints, which models the data attributes and gives semantic meaning to EHR data, while templates are graphical user interfaces created at runtime from the specifications defined in the archetypes [Pedersen et.al 2015].

The application of archetypes in the initial life cycle of a software system or in the integration with legacy HIS reveals several types of difficulties that must be addressed [Wang et.al 2015]. For example, the lack of tools that allow the generation and customization of graphical user interfaces with EHR data persistence functionalities, and the construction of data schemas using the data attributes, terminologies, and constraints defined in archetypes. In addition, scientific community manifests emphasize the importance of specifying computational solutions to support the software industry in the development of HIS using archetypes [Lin et.al 2016].

This article presents a computational tool to build HIS endowed with the characteristics of reusability and extensibility. Template4EHR[1] allows healthcare professionals to build applications from EHR specifications defined in archetypes, thus minimizing their dependence on a software development team. For this, the following HIS development resources allow Template4EHR to: i) dynamically generate relational and NoSQL data schemas; ii) create graphical user interfaces from the data attributes, terminologies and constraints of archetypes; iii) add and remove components of the generated graphical interface, even while it is running; iv) define the mandatory components of the graphical interface; and v) dynamically persist the data inserted in the graphical interface in different database systems.

2. Template4EHR Tool

This section presents the proposed tool for the generation of health applications using archetypes. Section 2.1 details the architecture and operation of Template4EHR, while Section 2.2 describes its main features.

2.1. Overview and Architecture

Template4EHR consists of a computational environment focused on the construction of data schemas and graphical user interfaces based on the EHR specifications of archetypes. The main purpose of the approach proposed here is the extraction of the data attributes, terminologies and constraints found in archetypes for EHR data storage in relational and NoSQL databases. Figure 1 shows a conceptual view of the operation of Template4EHR and the definition of each component is given below.

From the reading of an archetype imported by a health professional, Template4EHR performs the tasks of creating data schemas and generating graphical user interfaces. For this, the Extractor component shown in Figure 1 extracts from the imported archetype the attributes that define the EHR, the terminologies that give semantic meaning to clinical data, as well as the specified attributes constraints. After extraction, the archetype elements are stored in a document JSON (JavaScript Object Notation) for the generation of software artifacts by the Mapper component.

With the archetype elements in place, the Mapper component performs the following domain rules: i) maps data attributes to data entry components in the graphical user interface (e.g., text box, drop-down list); ii) uses the constraints extracted from the archetypes as a data entry validation mechanism in the graphical interface (e.g., range of values, constraint of data type); iii) provides the terminologies extracted from the archetypes in the graphical interface to give semantic meaning to their respective

[1] Template4EHR Demo: https://www.youtube.com/watch?v=Uoh-54ICxI8

components. After this activity is completed, the Generator component organizes the created components and then provides the graphical user interface.

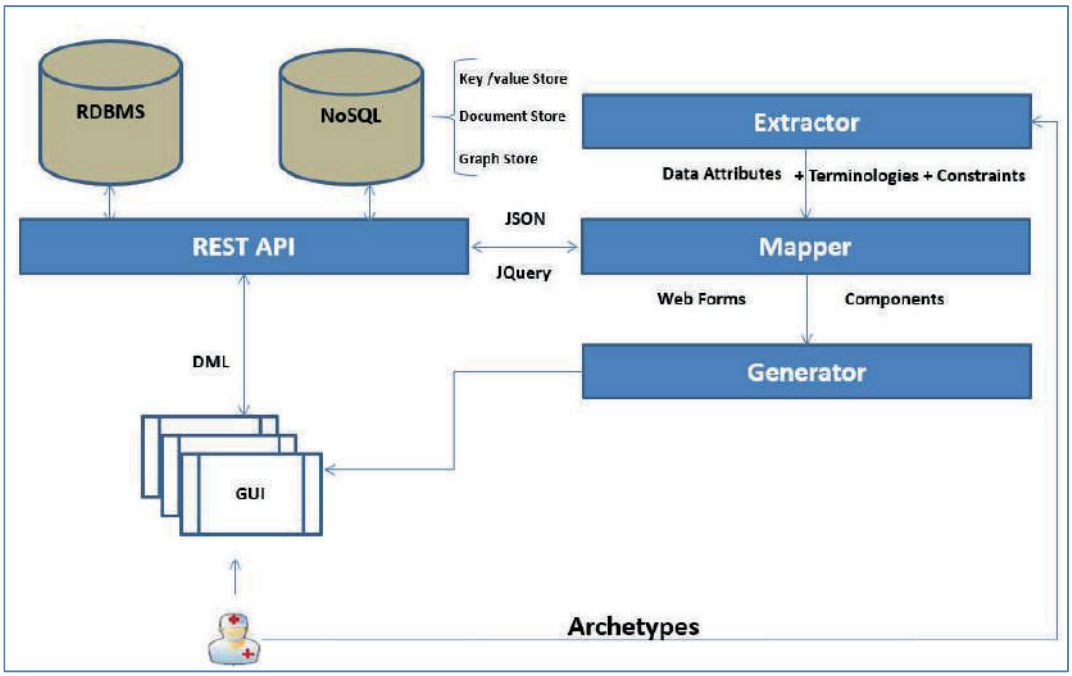


Figure 1. Template4EHR tool architecture

After the Extractor component extracts data attributes, terminologies, and constraints, the REST API / DCL generates relational and NoSQL data schemas. Template4EHR has a feature to configure the type of data schema to be generated when reading archetypes.

2.2. Main Features

In addition to generating graphical user interfaces and data schemas, Template4EHR has a set of features that allows end users to create a health application using archetypes. The definition of each one is given below.

Demographic information management: This functionality allows the management of health organizations, patients, physicians, nursing staff and users of an application instance created by Template4EHR. In a given health sector, an organization can be characterized as a hospital, clinic, laboratory, basic health unit, among others. In this context, Template4EHR creates application instances based on the organization type, such as an instance for the hospitalization of patients in a given hospital, and another instance focused on outpatient care in an emergency unit. In addition, patient, physician and nurse information managed by the tool can be added to the graphical user interfaces generated by Template4EHR.

Building Health Domains: An organization can offer a variety of health services to society. For example, a hospital may perform laboratory tests, diagnostic imaging, emergency care, hospitalization, etc. For this purpose, Template4EHR allows the creation and configuration of domains and subdomains that represent the services offered by each organization. In this case, the created domains and subdomains are used to organize and access the graphical interfaces generated by Template4EHR. This functionality can also

be used to create a sequence of activities to be performed by health professionals. For example, an application may have a domain called Nursing, in which professionals in this sector must perform activities such as vital signs evaluation and the physical examination of the patient.

Archetype Management: This feature imports and maps archetypes for a health application. When importing an archetype, Template4EHR allows the end user to choose which archetype data attributes will be part of the data schema and interface. Even after choosing the data attributes of the archetype, one can manage the generated graphical interface; adding, removing or disabling the attributes, causing Template4EHR to automatically extend the created data schema. Once this activity is complete, the graphical interface generated must be associated with an application domain / subdomain and the user access permissions granted. In addition, users are allowed to create their reports from the fields and information stored in the database.

Customizing Graphical User Interfaces: Template4EHR creates a standard graphical user interface from the reading of an archetype. To improve usability and user interaction, a set of features allow customization of the graphical interface. Among them, we highlight: i) modify or choose the type of components, for example, a text box can be configured as a single line or multiple lines; ii) modify the order of presentation of the components and iii) configure the fields that will be mandatory in the graphic interface.

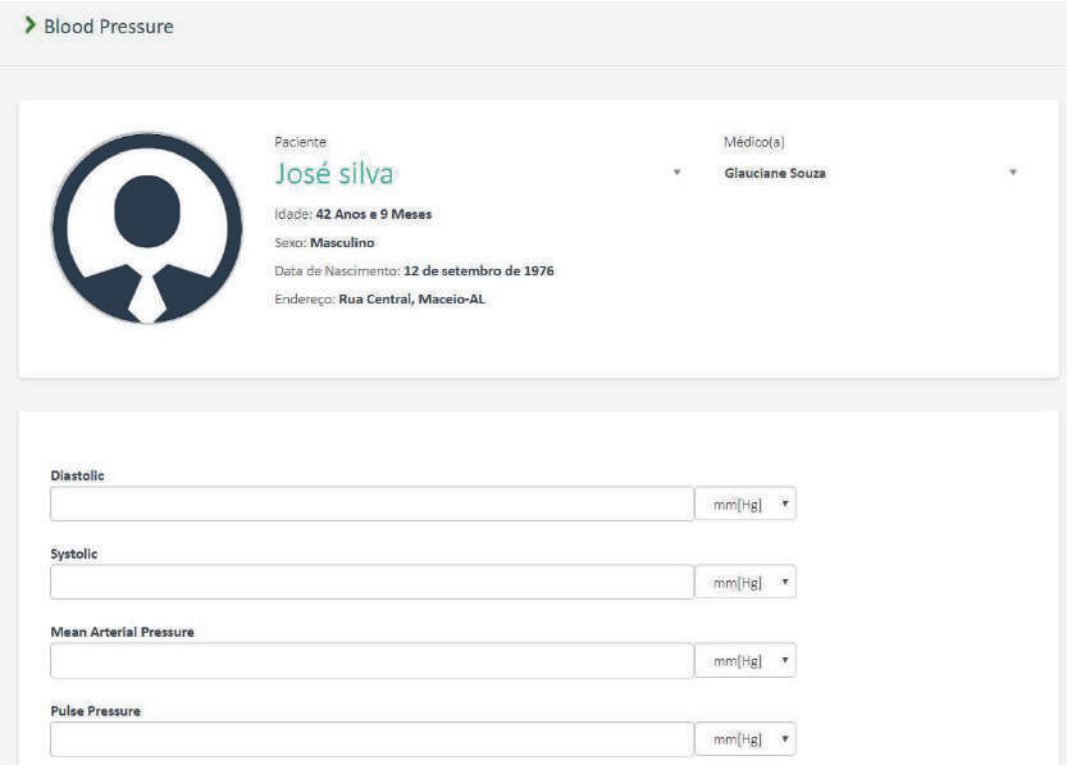


Figure 2. Graphical user interface generated from the blood pressure archetype

Figure 2 shows the graphical user interface created from reading the blood pressure archetype. Since the data attributes of the blood pressure archetype were specified as a list (i.e., ITEM_LIST), Template4EHR created the data schema in a

key-value database. Patient, medical and nursing information was stored in a relational database.

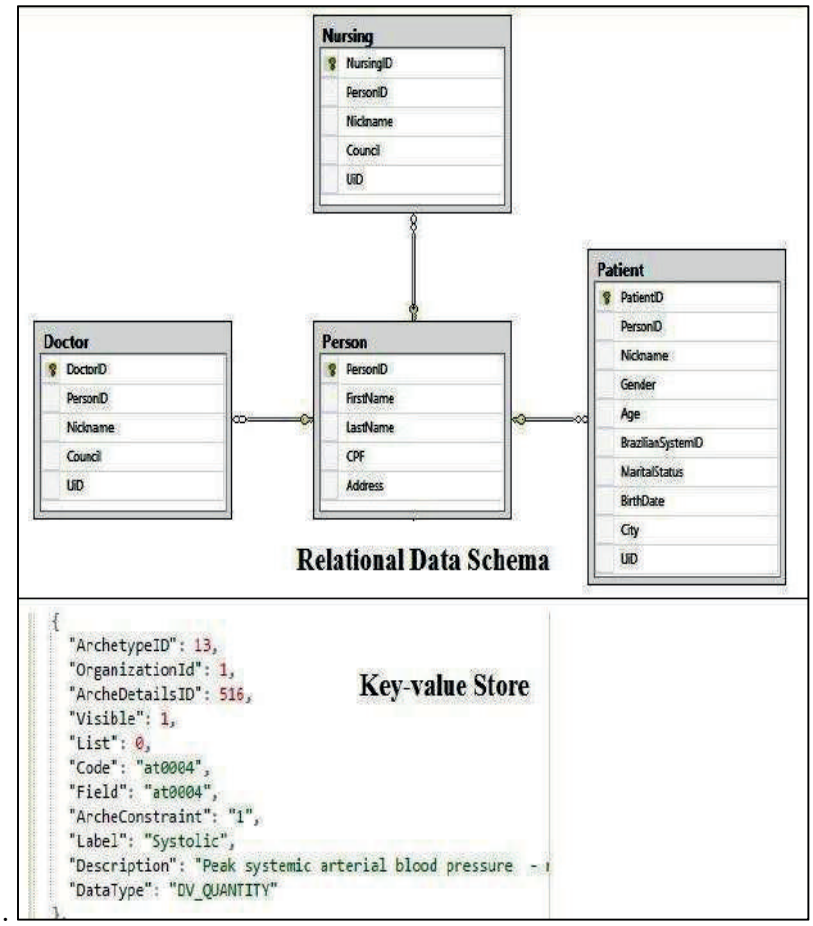


Figure 3. Data schema generated by the tool

Figure 3 shows parts of the relational and NoSQL data schema used in this example to store the data entered in the graphical interface generated from the blood pressure archetype.

3. Related Tools

Evaluating the solutions available on the market, we identified tools to generate GUIs using archetypes. EhrScape Framework (available at ehrscape.com) supports the health application development process using openEHR standard specifications. However, its GUI generation and customization resources are limited. For example, the inability to choose the data type for entry fields in the GUI. Cloud EHRServer from CaboLabs (cabolabs.com) features cloud services for EHR storage and dynamic building of GUIs that are compatible with the openEHR standard. Finally, the solution developed by EtherCIS (ethercis.org) offers a RESTful API compatible with archetypes, templates and archetype query language (AQL).

Although the works mentioned in this section represent a major advance in the state of the art, none of them allow the dynamic generation of data schemas in heterogeneous databases.

4. Conclusion

This article presents a computational tool capable of dynamically creating data schemas and graphical user interfaces from data attributes, terminologies and constraints specified in archetypes. The use of Template4EHR in the healthcare sector allows end users to build health applications based on the Electronic Health Record specifications defined in archetypes, thus minimizing their dependency on a software development team. The main features of the tool include: i) the generation of relational and NoSQL data schemas; ii) the generation and customization of graphical user interfaces and iii) the persistence of the data manipulated in the graphical interface into the data schemas created by the tool.

References

Georg D., Judith C., Christoph. R. (2013). Towards plug-and-play integration of archetypes into legacy electronic health record systems: the ArchiMed experience. BMC Medical Informatics and Decision Making. 2013; 1-12.

Jeffrey A. L., Jeffrey L. S, Blackford M. (2012) “Method of Electronic Health Record Documentation and quality of primary care”, J Am Med Inform Assoc, pp.1019-1024, 2012.

Jumaa H., Rubel P., Fayn. J. (2013) “An XML-based framework for automating data exchange in healthcare”, IEEE International Conference on e-Health Networking Applications and Services - Healthcom, pp.264 – 269, 2010

Lee K. k., Tangb W., Choia. K. (2013) “Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage”, Computer Methods and Programs in Biomedicine, pp. 99-109, 2013.

Lin CH, Fann YC, Liou DM. (2016) "An exploratory study using an openEHR 2-level modeling approach to represent common data elements."J Am Med Inform Assoc. 2016 Sep;23(5):956-67. doi: 10.1093/jamia/ocv137

Pedersen R, Wynn R, Ellingsen G. (2015) "Semantic Interoperable Electronic Patient Records: The Unfolding of Consensus based Archetypes."Stud Health Technol Inform. 2015; 210:170-4.

Wang L, Min L, Lu X, Duan H. (2015) “Archetype relational mapping - a practical openEHR persistence solution”; BMC Medical Informatics and Decision Making, pp. 1-18, 2015.

OndeBUS: uma Aplicação de Monitoramento e Detecção de Aglomerados de Ônibus

Lucas F. Oliveira[1], Demetrio G. Mestre[2], Veruska B. Santos[1], Andreza Raquel M. Queiroz[1], Carlos Eduardo S. Pires[1]

[1]Departamento de Sistemas e Computação
Universidade Federal de Campina Grande (UFCG)
Campina Grande, PB – Brasil

[2]Coordenação de Informática (CI)
Universidade Estadual da Paraíba (UEPB)
Campina Grande, PB – Brasil

lucas.oliveira@ccc.ufcg.edu.br, cesp@computacao.ufcg.edu.br

demetriogm@uepb.br, {veruska,andrezaraquel}@copin.ufcg.edu.br

Abstract. *One of the most recurrent problems with public transportation in Brazil is the bus bunching, an event consisting of two or more buses executing the same route together and that contribute to the increase in the waiting time of passengers at bus stops. In this work, we propose a bus fleet monitoring application[1] capable of detecting and displaying the occurrence of bus bunchings. To detect these bus bunchings we apply metrics based on the distance between buses in real time and in the time interval of arrival of buses in the stops. The tool aims to facilitate the decision making of the transport agents.*

Resumo. *Um dos problemas mais recorrentes com o transporte público no Brasil são os aglomerados de ônibus, evento constituído por dois ou mais ônibus executando a mesma rota juntos e que contribuem para o aumento no tempo de espera de passageiros em paradas. Desse modo, foi desenvolvida uma aplicação de monitoramento de frotas de ônibus capaz de detectar e exibir a ocorrência de aglomerados. Para detecção dos aglomerados, são utilizadas métricas baseadas na distância entre os ônibus em tempo real e no intervalo de tempo de chegada dos ônibus nas paradas. A ferramenta visa facilitar a tomada de decisão dos agentes de transporte.*

1. Introdução

Em um ambiente estocástico, como o trânsito nas cidades grandes, é comum acontecer atrasos no tempo de chegada dos ônibus nas paradas. Uma das principais preocupações das empresas de transporte é a formação de aglomerados de ônibus (em inglês, *bus bunching*), evento constituído por dois ou mais ônibus executando a mesma rota (trajeto programado) juntos. O problema dos aglomerados de ônibus está diretamente relacionado a fatores como horários de pico, acidentes de trânsito, poucos recursos para suprir grandes demandas e desvio do horário programado.

[1]Um vídeo de demonstração da ferramenta OndeBus pode ser encontrado em: https://youtu.be/EGobIOkU4Qw.

Por ser um problema cíclico, é de interesse das empresas de transporte evitar a ocorrência dos aglomerados. Um pequeno atraso ou congestionamento faz aumentar a quantidade de pessoas nas paradas de ônibus. Um ônibus, por sua vez, permanece mais tempo na parada pegando todos esses passageiros e, enquanto este transita superlotado, o próximo ônibus seguirá rápido e vazio. Eventualmente, um ônibus irá alcançar o outro, formando, assim, o aglomerado [Moreira-Matias et al. 2012] [Arriagada et al. 2019]. Esse cenário provoca um acréscimo no tempo de espera dos passageiros por um ônibus nas paradas seguintes e uma distribuição irregular nos veículos na rota, evidenciando um desperdício de recursos e refletindo um serviço sem credibilidade à população.

Nesse contexto, foi desenvolvida uma aplicação para monitoramento de frotas de ônibus, capaz de detectar e exibir no mapa da cidade, em tempo real, a ocorrência de problemas, como ônibus aglomerados e fora da rota. Com isso, espera-se que a tomada de decisão dos agentes de trânsito seja facilitada, através da exibição destes problemas, contribuindo com a melhoria da eficiência e confiabilidade do serviço de transporte público.

2. Trabalhos Relacionados

Em relação ao monitoramento de frotas de ônibus, a aplicação mais conhecida é o Google Maps[2], que exibe a localização das paradas e os horários que os ônibus irão chegar nelas, baseando-se em dados de GPS enviados em tempo real. A aplicação Olho Vivo[3] da SPTrans de São Paulo, além de informações sobre paradas e previsões dos horários de chegada dos ônibus, exibe as rotas, a localização dos veículos, a velocidade média e o tempo de percurso das vias. A aplicação CIOM-CG[4] de Campina Grande exibe apenas os ônibus executando suas rotas em tempo real, onde é possível filtrar por ônibus e ver os horários programados. No aplicativo para *smartphones* Cadê o Ônibus?[5], é possível calcular rotas, localizar as paradas mais próximas, consultar itinerários e acompanhar a localização dos ônibus de São Paulo em tempo real. No Moovit[6], aplicativo que dispõe de informações sobre linhas de ônibus, metrô, trem, barcas e teleféricos, é possível visualizar horários programados e os itinerários das linhas, além de comparar trajetos com informações sobre horários em tempo real.

Todas as aplicações têm em comum a disponibilização de informações programadas de ônibus e a localização em tempo real dos mesmos. Entretanto, como o foco dessas aplicações é o usuário de ônibus, o único problema detectado e informado por elas é o atraso dos horários de chegada nas paradas. Nenhuma delas facilita a tomada de decisão dos agentes de trânsito, porque, em geral, as aplicações são voltadas para as empresas privadas. Neste contexto, a aplicação aqui proposta diferencia-se das já publicadas ao realizar a detecção de problemas, como a ocorrência de aglomerados, ônibus fora da rota e não cumprimento dos horários programados, indicando, assim, o status dos veículos, em tempo real, em relação aos seus horários e rotas programados.

[2] https://www.google.com.br/maps
[3] http://olhovivo.sptrans.com.br/
[4] https://www.ciomcg.com.br/
[5] http://www.cadeoonibus.com.br/CoO/SiteV2
[6] https://moovitapp.com/

3. Aplicação OndeBUS

O OndeBUS é uma aplicação para auxiliar o monitoramento de frotas de ônibus das cidades, exibindo em um mapa a posição dos ônibus em tempo real. A aplicação consome dados de GPS (*Global Positioning System*) emitidos pelos ônibus e de GTFS (*General Transit Feed Specification*)[7]. Os dados de GPS contêm a geolocalização, o horário de envio do dado, o código e a rota do ônibus. Por sua vez, o GTFS, geralmente disponibilizado pelas empresas de ônibus, é um dado estático que fornece informações de localização das paradas, rotas, cronograma a ser seguido pelos ônibus, entre outras informações. Estes dados são pré-processados pela biblioteca BULMA Real Time (BULMA_RT) [Mestre 2018], que integra os dados de GPS e GTFS para associar rotas e status a cada ônibus.

A Figura 1 exibe o fluxo de execução da aplicação. Os dados de GTFS e de GPS emitidos pelos ônibus são pré-processados e integrados pelo componente BULMA_RT e, em seguida, lidos pelo componente 'OndeBUS', responsável por processar os dados lidos e realizar a detecção dos aglomerados de ônibus. A detecção é realizada com base nas métricas de distância e de *headway*, a serem descritas na Seção 4. Por fim, as informações são enviadas ao servidor, processadas pela interface e consumidas pelo agente de trânsito.

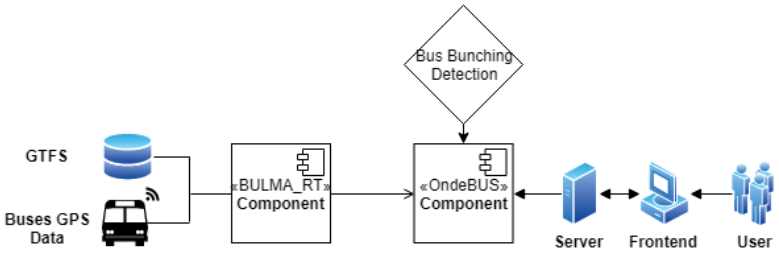


Figura 1. Diagrama de componentes da aplicação OndeBUS.

A Figura 2 ilustra a tela de monitoramento dos ônibus. Uma representação visual (baseada em cores) é utilizada para indicar o status dos ônibus, a citar: regular (ícone azul), atrasado (laranja), adiantado (verde), aglomerado (roxo) e fora da rota (vermelho). Esse status é determinado com base nas rotas e horários programados no GTFS. É também possível filtrar (Figura 3b) o conteúdo exibido no mapa por rota e código do ônibus para facilitar a visualização. Caso a opção *Show Bus Bunching* seja selecionada na tela (Figura 3a), os ônibus aglomerados passam a ser exibidos na cor roxa enquanto que os ônibus não aglomerados são exibidos na cor cinza. Além disso, é exibido na lateral um painel com informações sobre os ônibus aglomerados, como rota e código dos ônibus envolvidos.

4. Detecção de Aglomerados de Ônibus

A detecção automática de aglomerados de ônibus é feita a partir da utilização de métricas derivadas do *headway* e da distância entre pares de ônibus executando a mesma rota. Estas métricas servem para as empresas avaliarem a ocorrência dos aglomerados com base em dois pontos de vista: proximidade física e horário programado.

4.1. Detecção de Aglomerados por Headway

O *headway* (h) é o intervalo de tempo de chegada de dois ônibus da mesma rota em uma mesma parada [Feng and Figliozzi 2011], medido em segundos neste trabalho. Os autores

[7] https://developers.google.com/transit/gtfs

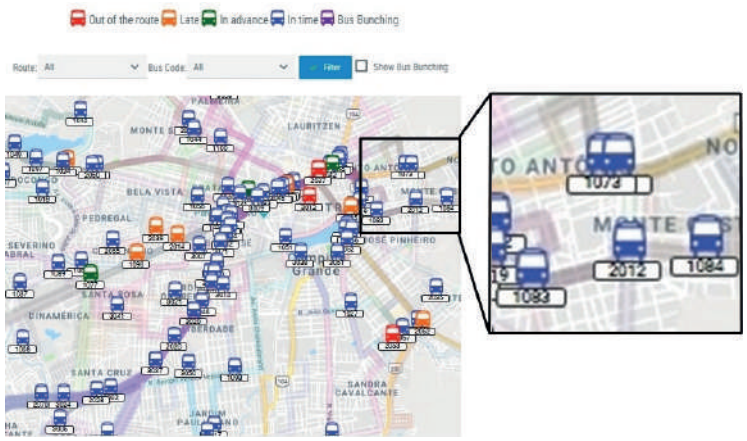


Figura 2. Visualização dos ônibus em tempo real na aplicação OndeBUS.

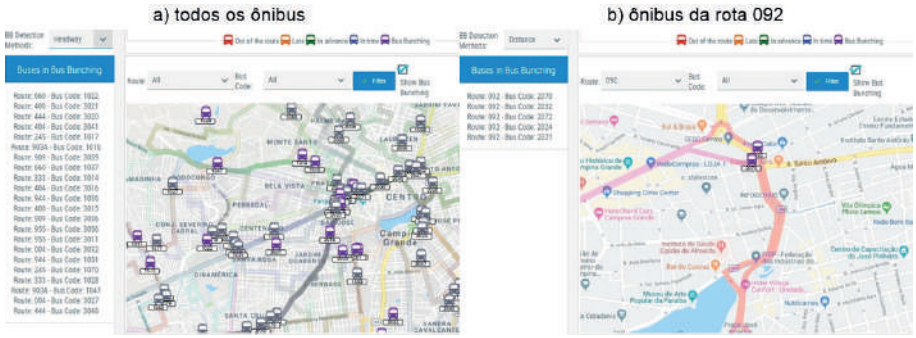


Figura 3. Visualização dos ônibus aglomerados no OndeBUS: a) todos os ônibus (sem filtro) e b) apenas os ônibus da rota 092.

de [Arriagada et al. 2019] e [Yu et al. 2016] constataram que os aglomerados de ônibus acontecem devido à instabilidade de *headways*, ou seja, quando o $h_{observado}$ é próximo de zero ou menor do que uma porcentagem do $h_{programado}$. O $h_{observado}$ é o *headway* que ocorre na prática, enquanto o $h_{programado}$ é o *headway* planejado que os ônibus deveriam obedecer. De acordo com as definições encontradas em [Arriagada et al. 2019], um aglomerado de ônibus ocorre quando:

$$h_{observado} < \frac{h_{programado}}{4} \tag{1}$$

ou seja, quando o $h_{observado}$ é menor que 25% do $h_{programado}$, tendo a parada de ônibus como ponto de referência. O fracionamento de 25% foi sugerido pelos autores de [Arriagada et al. 2019] para aumentar o intervalo de confiança nas detecções e evitar possíveis resultados falsos positivos.

4.2. Detecção de Aglomerados por Distância

A métrica de *headway* detecta aglomerados apenas quando os ônibus chegam em uma parada. Entretanto, em algumas rotas, as paradas de ônibus estão localizadas a uma distância significativa umas das outras. Nesse caso, se um aglomerado acontecer durante o momento em que os ônibus estão trafegando entre as paradas, o mesmo não será detectado pela métrica *headway*. Assim, faz-se necessária uma métrica que considere a distância física entre os ônibus.

Em um cenário ideal, os ônibus de uma mesma rota devem estar uniformemente distribuídos em termos de distância uns dos outros. A distribuição pode ocorrer com base nos seguintes fatores: tamanho da rota e quantidade de ônibus executando a rota. Como não foram encontrados trabalhos relacionados que usam a distância entre os ônibus em tempo real como parâmetro para a detecção de aglomerados, foi proposta a seguinte métrica de distância:

$$distância_{ab} < \frac{tamanho\ da\ rota/n^{o}\ de\ ônibus}{4} \quad (2)$$

De acordo com a Equação 2, um aglomerado ocorre quando a distância entre dois ônibus A e B, determinada pelos dados de GPS a e b, respectivamente, é menor que 25% do tamanho da rota dividido pela quantidade de ônibus que a executam. Para evitar comparações entre um dado de GPS atual e um antigo, a comparação da distância entre dois ônibus só é feita se a diferença entre o tempo de envio dos dados a e b for de no máximo t segundos, neste caso, foi utilizado $t = 60$.

4.3. Aplicação das Métricas

A seguir, são apresentados resultados obtidos com a aplicação das métricas apresentadas. Para tal, foram utilizados dados de GPS dos ônibus[8] da cidade de Campina Grande do dia 21 de outubro de 2017 no horário de 11h às 14h (horário de pico). A Figura 4a exibe um histograma de ocorrências de aglomerados de ônibus detectados pela métrica de *headway*. É possível observar que mais de 75% dos pares de ônibus aglomerados apresentam $h_{observado}$ entre [0, 75] segundos. Para um $h_{observado} > 125$, verifica-se que há poucas ocorrências de aglomerados, sendo $h_{observado} = 300$ o máximo encontrado. Os resultados confirmam uma detecção eficaz dos aglomerados de ônibus, pois, como citado na Seção 4.1, este evento acontece quando o *headway* está próximo de zero.

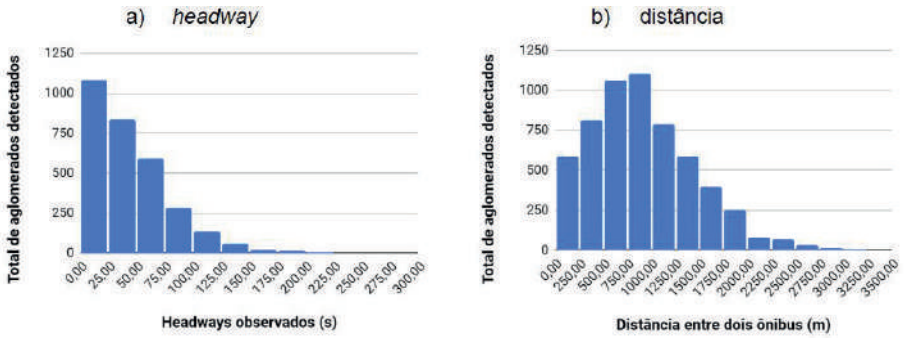


Figura 4. Histograma da ocorrência de aglomerados detectados com base na métrica de: *a) headway* e *b) distância*.

Em relação à detecção de aglomerados de ônibus por distância (Figura 4b), os resultados apresentam-se mais dispersos no histograma, porém com concentração também à esquerda, onde as distâncias entre os pares de ônibus são menores. A maior parte das detecções ocorrem quando os pares de ônibus estão entre 0 e 2.000 metros de distância entre si. Apesar de se encontrarem dentro do limiar de distância estabelecido (Equação 2), os resultados mais à direita do gráfico não fornecem garantia de que os pares de ônibus

[8]O envio de dados em tempo real foi simulado.

realmente estejam aglomerados. A detecção desses casos acontece, em geral, em rotas de grande extensão servidas por poucos ônibus. Conclui-se que quanto maior o tamanho da rota, maior o espaçamento entre os ônibus, refletindo diretamente no aumento do limiar.

5. Conclusões e Trabalhos Futuros

Nesta demonstração, foi apresentada uma aplicação de monitoramento de frotas de ônibus capaz de detectar aglomerados de ônibus. Utilizando a métrica baseada no *headway*, foi possível identificar aglomerados de forma eficaz, em geral de 0 a 75 segundos, de acordo com os dados analisados. Utilizando a métrica de distância, também foi possível identificar aglomerados de forma eficaz, em geral de 0 a 1.000 metros. Porém, foram detectados alguns ônibus aglomerados com distância de até 3.500 metros, o que contradiz a definição de aglomerado de ônibus [Feng and Figliozzi 2011].

Como trabalhos futuros, espera-se disponibilizar relatórios acerca do desempenho das frotas de ônibus e torná-la responsiva, para facilitar o acesso dos agentes de trânsito por *smartphones*. Espera-se também melhorar a definição da métrica de distância considerando outros fatores, como a frequência da rota, com o intuito de deixá-la mais adequada à definição de aglomerados de ônibus e minimizar o aparecimento de casos falsos positivos. Além disso, visando melhorar a precisão dos resultados da aplicação, pretende-se implementar um método de detecção híbrido, ou seja, que considere as métricas de distância e *headway* simultaneamente. Finalmente, pretende-se testar a aplicação com dados de outras cidades e coletados em tempo real.

Agradecimentos

Esta pesquisa foi parcialmente financiada pelo INES 2.0, concessão da FACEPE APQ-0399-1.03/17, concessão da CAPES 88887.136410/2017-00 e concessão do CNPq 465614/2014-0.

Referências

Arriagada, J., Gschwender, A., Munizaga, M. A., and Trépanier, M. (2019). Modeling bus bunching using massive location and fare collection data. *Journal of Intelligent Transportation Systems*, 23(4):332–344.

Feng, W. and Figliozzi, M. (2011). Using archived avl/apc bus data to identify spatial-temporal causes of bus bunching. In *90th Annual Meeting of the Transportation Research Board*.

Mestre, D. G. (2018). *Leveraging the entity matching performance through adaptive indexing and efficient parallelization*. PhD thesis, Universidade Federal de Campina Grande.

Moreira-Matias, L., Ferreira, C., Gama, J., Mendes-Moreira, J., and De Sousa, J. F. (2012). Bus bunching detection: A sequence mining approach. In *Workshop on Ubiquitous Data Mining*, page 13.

Yu, H., Chen, D., Wu, Z., Ma, X., and Wang, Y. (2016). Headway-based bus bunching prediction using transit smart card data. *Transportation Research Part C: Emerging Technologies*, 72:45–59.

J-EDA: A diversified similarity workbench for content-based image retrieval*

João V. O. Novaes[1], Marcos V. N. Bedo[2], Daniel de Oliveira[3], Agma J. M. Traina[1], Caetano Traina Jr.[1], and Lúcio F. D. Santos[4]

[1] Institute of Mathematics and Computer Science – University of São Paulo (ICMC/USP)
São Carlos/SP -– Brazil

[2]Fluminense Northwest Institute – Fluminense Federal University (INFES/UFF)
St. A. Pádua/RJ – Brazil

[3]Institute of Computing – Fluminense Federal University (IC/UFF)
Niterói/RJ – Brazil

[4]Federal Institute of Technology of North of Minas Gerais (IFNMG)
Montes Claros - MG - Brazil

novaes.jvo@usp.br, {agma,caetano}@icmc.usp.br, marcosbedo@id.uff.br,
danielcmo@ic.uff.br, lucio.santos@ifnmg.edu.br

Abstract. *Similarity searching is employed for content-based image retrieval (CBIR) as a fast and explainable query mechanism. However, standard similarity searches may be unsuitable for querying large data sources as retrieved elements are prone to be very similar among themselves. While adding diversity into similarity searching enhances the result set semantics in such cases, the fine tuning of query settings must be performed through experimental evaluations This paper introduces J-EDA[1], a practical workbench implemented in JAVA for the analysis of diversified similarity queries regarding content-based image retrieval tasks. J-EDA includes a broad variety of search methods in the literature as well as several query parameters for the execution of content-based image retrieval of supervised and unsupervised data sources, such as distance functions, image feature extractors, query criteria and relevance feedback techniques. Moreover, J-EDA provides a set of internal and external metrics, e.g., $P \times R$ and Mean Average Precision (mAP), which are reported at the end of experiments. Such quality metrics can be oriented towards either incremental or batch procedures and with and without human interaction.*

1. Introduction

Content-Based Image Retrieval (CBIR) systems are prime tools for recovering images that are the most similar to a given query reference (or *query image*) regarding their visual content rather than additional tags or labels. Such tools rely on distance functions for calculating the (dis) similarity between visual features extracted from the query reference and other images from a database. Accordingly, CBIRs typically include a *description*

*This study was financed in part by the CAPES - Finance Code 001, FAPEMIG, FAPERJ (E. Sediadas/2018), FAPESP (Project nº 2016/17078-0) and CNPq

[1]Available at: github.com/NovaesJVO/J-EDA

phase where each image is represented as a feature vector that summarizes visual contents collected by either image processing-driven transformations, or machine learning-based approaches. The feature vectors of stored images and that of query elements are compared by distance functions according to criteria borrowed from the Metric Spaces theory [Zezula et al. 2010]. Two of the most employed distance-based operators are the Range (R_q) and the k-Nearest Neighbor (k-NN$_q$) queries: the former retrieves the images whose distances to the query image is not greater than a maximum radius ξ, while the latter recovers the k closest images to the query reference [Bedo and et. al 2016].

However, the problem with CBIR systems is that they are designed and adjusted for specific domains as their parameters profoundly impact the results of content-based searches [Bedo and et. al 2016]. Moreover, the querying of massive image databases may reduce the semantics within the result sets as standard distance-based operators disregard the handling of images that are too similar among themselves [Drosou et al. 2017]. Such a lack of result set expressiveness leads to the CBIR semantic gap where the query answer may contain nearly-duplicate information, which prevents the exploration of relevant portions of the search space and blurs decision-making routines. Bridging such semantic gap can be achieved through either user-driven relevance feedback and diversity induction [Santos and et. al 2018, Spyromitros and et. al 2015]. Relevance feedback enables shifting the query perspective, or even the query reference, according to user-provided annotations over the result set, while diversity induction aims at producing a result set with elements not only similar to the query element but also diversified among themselves.

Therefore, the fine-tuning of CBIR systems requires the correct choice of *search parameters* according to data and characteristics of the domain at hand. In particular, finding the following CBIR search parameters can be seamlessly addressed experimentally as data validations on an external workbench tool: *(i)* feature extraction, which determines how images are represented into a multidimensional space, *(ii)* distance function, which defines how the similarity between images is calculated, *(iii)* query criteria, which rules how the result sets are formed according to the similarity scores and distance-based operator, *(iv)* result set diversification, which regulates the level of similarity among the result images, and *(v)* relevance feedback, which enables the user to explore other perspectives and images in the search space. Although the literature reports workbench prototypes for the evaluation search parameters, they are focused on specific parameters and are unable to provide metrics regarding the tuning of all five parameters, simultaneously.

For instance, `DivDB` [Vieira and et. al 2011] enables the comparison of several diversity-based algorithms as well as their thresholds but does not provide an evaluation of those methods alongside with relevance feedback approaches. Likewise, CBIR `ImageHunter` [Tronci and et. al 2013] addresses the evaluation of several relevance feedback approaches but does not provide support for result set diversification. `Poikilo` [Drosou and Pitoura 2013] enables the evaluation of diversity-based algorithms with and without fixed query criteria and relevance feedback, distance functions and image representation are not considered as part of search tuning. A final prototype, `VikS` [Santos and et. al 2014], inspects distance functions and feature extraction, but the impact of such parameters are expressed by data visualization-only techniques.

This demonstration overcomes those limitations in the implementation of J-Environment for Diversity Analysis (J-EDA), a practical JAVA workbench for the analy-

sis of the impact of five content-based image retrieval parameters. J-EDA enables users to provide image datasets to be either processed feature extractors or to be associated with multidimensional representations provided in a separated `.csv` file. Next, users can inspect a broad set of search parameters as distance functions, query criteria, result diversification, and relevance feedback methods. Data evaluation is carried out through an Experimental interface or by an incremental batch procedure that generates distinct metrics for supervised and unsupervised datasets. Consolidated metrics, including elapsed time, are reported in a separated interface with easy-to-follow graphics and tables.

2. The J-EDA Workbench

J-EDA workbench is a JAVA desktop application implemented according to the modular architecture of Figure 1. Roughly speaking, J-EDA consists of eight main components, namely: *(i)* Graphical User Interface (GUI), *(ii)* Controler, *(iii)* Feature Extractor, *(iv)* Distance Evaluator, *(v)* Data Handler, *(vi)* Retrieval Module, *(vii)* Relevance Feedback Module, and *(viii)* Evaluation Metrics Module. J-EDA GUI enables users to set input parameters as well as choose the experimental setup to be evaluated, whereas the Controller translates the user inputs into commands for the implementation modules. The Feature Extractor component provides an implementation of classical low-level algorithms for describing images regarding color, texture, and shape.

The Distance Evaluator component implements the following set of weighted distance functions Bray-Curtis, Canberra, L_∞, L_2 and L_1. In its turn, the Data Handler component provides a common data structure for pairing images and feature vectors for content-based retrieval. J-EDA Retrieval module provides both similarity query criteria k-NN$_q$and $R_q$that can be combined with the following result set diversification methods: *(i)* Better Result with Influence Diversification (BRID), *(ii)* First Match (FM), *(iii)* Greedy Marginal Contribution (GMC), *(iv)* Maximal Marginal Relevance (MMR), and *(v)* Relative Grouping based on Influence (ReGi). Analogously, the Relevance Feedback module includes the following user-driven query feedback approaches: *(i)* Query Point Movement, *(ii)* Support Vector Machines-based Relevance, and *(iii)* Similarity Refinement. A theoretical background regarding the implemented result diversification and relevance feedback approaches can be found at J-EDA repository[1]. The system final module is the

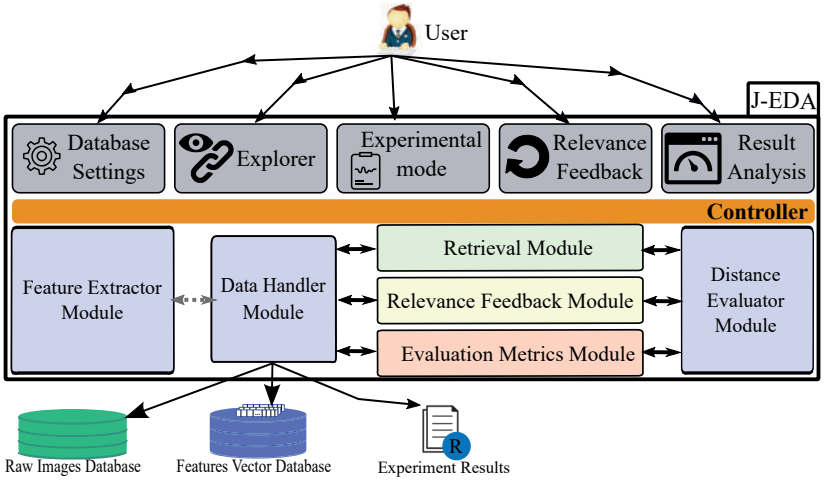


Figure 1. J-EDA architecture and components.

Evaluation Metrics that quantify the experimental trials and reports quality measures such as Precision and Recall ($P \times R$) and Mean Average Precision (mAP).

3. Demonstration of J-EDA Workbench

Here, we present a demonstration on J-EDA by using the well-known labeled dataset `Corel`[2]. The dataset includes $1,000$ images whose contents are classified into 10 mutually exclusive labels. Upon entering J-EDA, users are provided with the main workbench interface – Figure 2. It includes four upper tabs that summarize the experimental types available on J-EDA, namely (1) Explorer, which enables the manual evaluation of a single query image regarding the evaluated dataset, (2) Experimental batch mode with supervised feedback, which enables the batch evaluation of image datasets with user-interactions, (3) Experimental batch mode with unsupervised feedback, which automatically evaluates query parameters on labeled image datasets, and (4) Result set analysis, which provides the metrics for the evaluations performed in the three previous tabs.

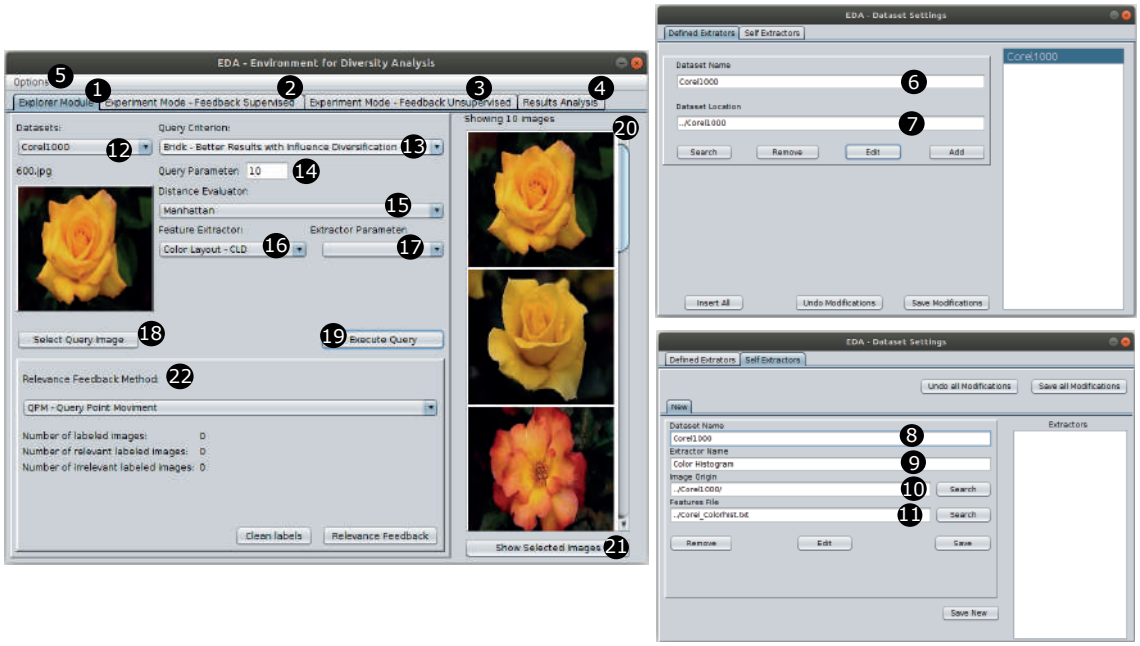


Figure 2. J-EDA main interface with Explorer mode and query setup.

Both query setting and data parameters can be defined as options (5) in the J-EDA main interface, whereas users can either set J-EDA to extract visual features regarding the original set of images and labels (6–7), or provide feature vectors produced by other tools, *e.g.*, features drawn from CNN autoencoders, as a separated `.csv` file. In the former case, users must inform both dataset and external extractor name (8–9), image data directory (10), and the file location of the features and labels (11). Back to the main Explorer panel, users may choose the query parameters, such as the similarity query criteria and its values (13–14), *e.g.*, number of neighbors, the distance function (15) as well as the feature extractor name and its values (16–17), *e.g.*, number of color histogram bins. Finally, users must provide a query reference (18) and run the similarity query (19). The panel displayed on (20) shows the result set for the setup of Figure 2 over `Corel`. Users may visualize

[2]Available at `http://wang.ist.psu.edu/docs/related/`

every result in full scale (21) as well as label each image as relevant or non-relevant by using right and left clicks, respectively. A relevance cycle is performed by using both the dynamic labels and a user-defined feedback method (22).

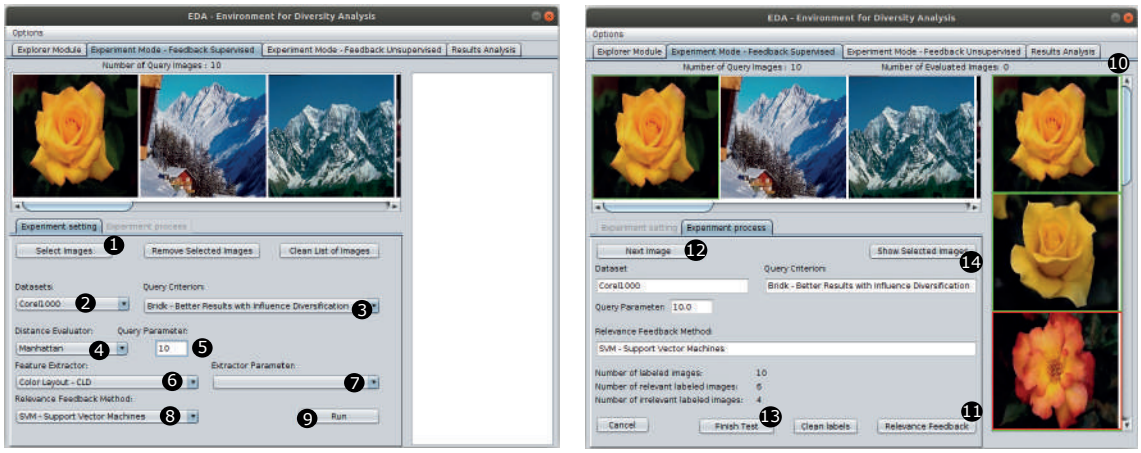


Figure 3. Supervised feedback support on J-EDA Experimental GUI.

The Experimental mode GUI automates the inspection of several query parameters – Figure 3. The GUI is specialized into two distinct versions regarding supervised feedback and unsupervised feedback. The former focuses on human-in-the-loops interactions in which users are requested to label (a sample of) retrieved images before the execution of relevance feedback cycles. Such interactions assume the dataset images are not related to class annotation, being particularly suitable for *unsupervised* datasets. The latter GUI specialization performs the automatic assessment of relevant images based on the label associated with each result set element – a task applicable to *supervised* datasets.

Figure 3 presents the demonstration setup for the *holdout* evaluation of a particular query setup over `Corel` (2–8). In this case, 10% of the images were detached from `Corel` as a stratified sample and stored into a separated directory (1) to be queried in batch mode (9). J-EDA shows the retrieved images for every reference into a separated panel (10), where users inspect the results (14) and label them as relevant (green) or non-relevant (red). Feedback cycles are executed upon request and as long as necessary (11). Next, users can either proceed to the next image (12) or interrupt the batch assessment (13).

Analogously, Figure 4(a) presents `Corel` batch evaluation with unsupervised feedback. In this case, relevance is inferred upon the classes of both query and result images, *i.e.*, equal labels imply into relevant results. Besides the query parameters (1) and the number of feedback cycles (2), users must provide the set of images as either *(i)* a random stratified sample of a repository of images (4), *(ii)* a script containing the set of query reference locations (5), or *(iii)* a manual choice of images in the file system (6). Figure 4(a) shows the running of a stratified random sample of 10% (without replacement) of `Corel` dataset (7), whereas the experimental log is stored into an external file.

As the last part of the demonstration, Figure 4(b) presents the easy-to-follow interface of Results Analysis GUI with the mAP metric obtained from two query setups of the experimental trial in Figure 4(a). In the interface, users can choose several supervised and unsupervised metrics (10) and plot types to analyze pieces of results (11) recovered from log files (8–9). The produced plots can also be stored for further inspections (12).

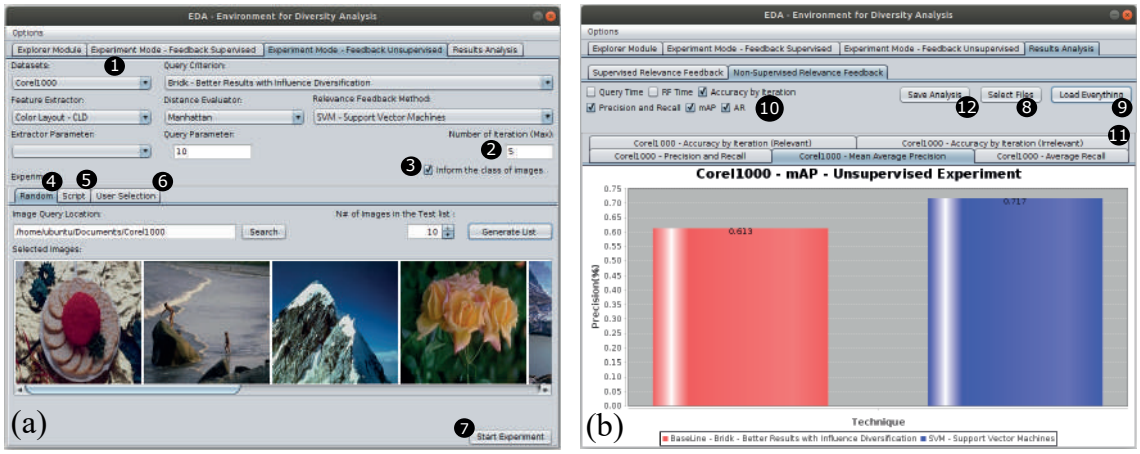


Figure 4. (a) Experimental mode GUI with unsupervised feedback for labeled datasets. (b) Result Analysis main interface.

4. Conclusions

This demonstration introduced the J-EDA workbench, an easy to use and powerful environment for determining suitable settings for querying generic datasets by similarity. Besides the Explorer, J-EDA provides two experimentation modules that benefit from both human-in-the-loop interactions as well as automatic assessments of labeled datasets. Lastly, J-EDA provides an analytic module that allows users to explore metrics and plots regarding distinct query parameters.

References

Bedo, M. and et. al (2016). Endowing a content-based medical image retrieval system with perceptual similarity using ensemble strategy. *JDI*, 29(1):22–37.

Drosou, M., Jagadish, H., Pitoura, E., and Stoyanovich, J. (2017). Diversity in big data: A review. *Big data*, 5(2):73–84.

Drosou, M. and Pitoura, E. (2013). POIKILO: A tool for evaluating the results of diversification models and algorithms. *PVLDB*, 6(12):1246–1249.

Santos, L. and et. al (2014). Have you met VikS?: A novel framework for visual diversity search analysis. In *SBBD Demos*, pages 209–214. SBC.

Santos, L. and et. al (2018). Exploring Diversified Similarity with Kundaha. In *CIKM*, pages 1903–1906. ACM.

Spyromitros, E. and et. al (2015). Improving diversity in image search via supervised relevance scoring. In *ICMR*, pages 323–330.

Tronci, R. and et. al (2013). ImageHunter: A Novel Tool for Relevance Feedback in Content Based Image Retrieval. In *DART*, pages 53–70. Springer.

Vieira, M. and et. al (2011). DivDB: A System for Diversifying Query Results. *PVLDB*, 4(12):1395–1398.

Zezula, P., Amato, G., Dohnal, V., and Batko, M. (2010). *Similarity Search: The Metric Space Approach*, volume 2. Springer.

Iago: um Sistema Gerenciador de Dados na Web

Wilker Cavalcante do Rego Santos[1], Lairson Emanuel R. de Alencar Oliveira[1], Thiago Moura da Silva[1], Marcelo Iury S. Oliveira[2], Bernadette Farias Lóscio[1]

[1]Centro de Informática -– Universidade Federal de Pernambuco (UFPE)

[2]Unidade Acadêmica de Serra Talhada – Universidade Federal Rural de Pernambuco

{wcrs,lerao,tms3,bfl}@cin.ufpe.br, marcelo.iury@ufrpe.br

Resumo. *A Web tornou-se uma importante ferramenta de compartilhamento e consumo de dados e informações. Apesar da ideia de compartilhamento de dados na Web não ser nova, inúmeros desafios ainda podem ser encontrados e estes podem dificultar a comunicação entre quem compartilha os dados e quem os consome, como o gerenciamento dos metadados, atualização dos conjuntos de dados e versionamento dos mesmos. Neste contexto, este trabalho propõe o Iago, um Sistema Gerenciador de Dados na Web, que, além de seguir a maioria das boas práticas para dados na Web propostas pelo W3C, tem a finalidade de automatizar os processos envolvidos na publicação de conjuntos de dados na Web.*

1. Introdução

A Web tornou-se uma importante ferramenta de compartilhamento e consumo de dados e informações. Com sua popularização, a quantidade de dados gerados por seus usuários cresce a cada dia. Em paralelo, novos paradigmas vem sendo desenvolvidos ao longo dos últimos anos com o intuito de descobrir formas inovadoras de utilização da Web.

Atualmente, existem diversas soluções para publicação de dados na Web, dentre as quais destacamos o CKAN[1], Socrata[2], Junar[3] e OpenDataSoft[4]. Apesar da sua ampla utilização, essas soluções não oferecem mecanismos para o gerenciamento adequado dos conjuntos de dados. Como limitações dessas soluções, podemos destacar a falta de mecanismos que possibilitem o gerenciamento de versões, atualização automática dos conjuntos de dados com frequências de atualização pré-definidas e gerenciamento de *feedback*.

Neste contexto, este trabalho tem como objetivo propor o Iago, um Sistema Gerenciador de Dados na Web, que oferece uma gama de serviços capazes de realizar tanto as tarefas já realizadas pelas soluções existentes de publicação de dados quanto as tarefas relacionadas ao gerenciamento de dados na Web. Ao implementar o Iago, consideramos a recomendação Data on the Web Best Practices [Lóscio et al. 2017], desenvolvida pelo W3C[5] (World Wide Web Consortium), que estabelece um conjunto de 35 boas práticas relacionadas a diferentes aspectos de publicação e consumo de dados na Web, como formatos de dados, acesso a dados, identificadores de dados e metadados.

[1]http://ckan.org
[2]http://socrata.com
[3]http://www.junar.com/
[4]https://www.opendatasoft.com/
[5]https://www.w3.org/Consortium/

O restante deste artigo está estruturado como se segue. Na Seção 2, é apresentado o Iago. Na Seção 3, é apresentada uma breve demonstração do funcionamento da ferramenta. Na Seção 4, são apresentadas as considerações finais.

2. Visão Geral do Iago

Segundo [Oliveira et al. 2018], um Sistema Gerenciador de Dados na Web (SGDW) pode ser caracterizado como um conjunto de serviços para compartilhamento de dados na Web, os quais facilitam a criação, manutenção, manipulação e consumo dos dados. A ideia de termos um SGDW surgiu com o propósito de preencher as lacunas das atuais soluções para compartilhamento de dados na Web. Neste contexto, propomos um SGDW, denominado, Iago.

O conjunto de serviços provido pelo Iago compreende uma solução que torna os dados acessíveis e utilizáveis tanto por seres humanos quanto por máquinas. Em particular, o Iago fornece também um conjunto de interfaces projetadas para simplificar a publicação, o compartilhamento, a descoberta e a utilização de dados, incluindo o armazenamento de dados e o provimento de APIs de dados. Desta forma, o Iago pode ser usado para rápida criação de portais de compartilhamento de dados na Web.

A Figura 1B apresenta uma análise comparativa entre o Iago e as principais soluções de publicação de dados mais citadas na literatura [Oliveira et al. 2017]. Como podemos observar, apesar de serem utilizadas para criação de diversos portais, as soluções atuais apresentam lacunas quanto aos requisitos necessários para uma solução de compartilhamento de dados na Web, como o gerenciamento de versões, curadoria de metadados, comunicação com os consumidores e produtores, e a preservação dos conjuntos de dados. O Iago, até onde sabemos, é a única ferramenta a dar suporte a todas estas características.

Dentre os principais diferenciais do Iago destacam-se: i) extração de dados de diversos tipos de fontes de dados; ii) geração automática de descrição dos Conjuntos de Dados por meio de metadados, legível por seres humanos e máquinas; iii) atualização automática dos conjuntos de dados de acordo com frequências de atualização pré-definidas; iv) gerenciamento de versões dos conjuntos de dados; v) geração automática de APIs para os conjuntos de dados criados; vi) geração automática de múltiplas distribuições para um mesmo conjunto de dados; vii) e descrição do catálogo de conjuntos de dados utilizando os vocabulários schema.org[6] e DCAT[7].

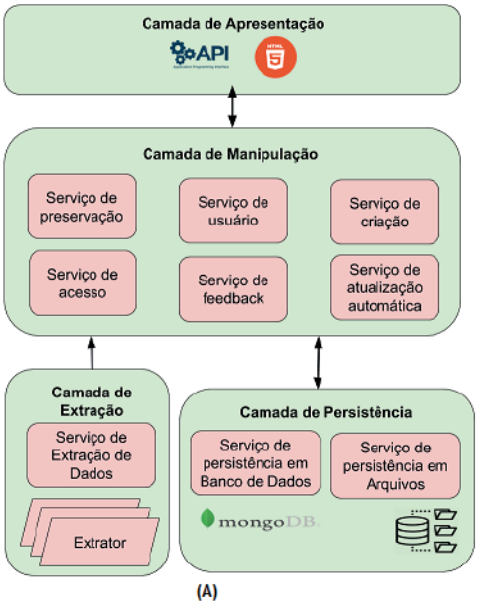
Uma visão geral da arquitetura de componentes do Iago é apresentada na Figura 1A. A fim de facilitar a comunicação entre tais componentes e preservar a independência de cada um deles, optamos por uma arquitetura baseada em serviços. Assim, caso necessário, os serviços podem ser substituídos (ou mantidos) de forma compartimentalizada, evitando problemas de acoplamento. No mais, os serviços são organizados em quatro camadas, assim como proposto em [Oliveira et al. 2018], permitindo uma melhor organização do projeto e a fácil manutenibilidade de seu código fonte.

2.1. Camada de Apresentação

A Camada de Apresentação é responsável pela interação dos usuários com o sistema. Nesta camada, são encontradas a interface Web e são geradas as APIs em uma interface

[6]https://schema.org/

[7]https://www.w3.org/TR/vocab-dcat/



(A)

	CKAN	SOCRATA	Junar	OpenDataSoft	IAGO
Extração semi-automática de dados	N	AP	AP	AP	A
Geração automática de metadados	AP	AP	AP	AP	AP
Atualização automática dos conjuntos de dados	N	A	N	AP	A
Gerenciamento de versões dos conjuntos de dados	AP	N	AP	N	A
Múltiplos mecanismos de acesso aos dados	A	A	A	A	A
Geração automática de múltiplas distribuições	N	A	N	A	A
Uso de vocabulários	A	A	A	A	A
Preservação dos conjutos de dados	N	N	N	N	A

(B)

Figura 1. (A) Arquitetura do Iago; (B) Quadro Comparativo de Soluções de Publicação de Dados[8]

remota para o acesso aos Conjuntos de Dados. Também é de responsabilidade da Camada de Apresentação, a validação de todos os valores de entrada no sistema, com o objetivo de evitar o acionamento de erros não esperados.

2.2. Camada de Manipulação

A Camada de Manipulação é responsável pelo gerenciamento dos dados e metadados, bem como pela implementação dos requisitos funcionais do sistema. O **Serviço de Acesso** é responsável por interpretar a requisição de acesso aos dados e retornar os dados nos termos desejados. Este serviço leva em consideração a versão, o formato de dados e se o usuário deseja retornar o conjunto de dados por completo ou não (retornar um subconjunto ou o Conjunto de Dados de forma paginada). O **Serviço de Usuário** é responsável pelas tarefas de autenticação e autorização do sistema, podendo bloquear acesso a outros serviços.

O **Serviço de Criação** é responsável pela criação de conjuntos de dados, dando início ao processo de publicação de conjunto de dados. Este serviço se destaca por ser o que mais gera dependências no sistema, fazendo dele uma peça central para o funcionamento do Iago. Nele, os metadados são organizados e os conjuntos de dados são automaticamente configurados para receber URIs únicas.

Por sua vez, o **Serviço de Atualização Automática** é responsável por coordenar a atualização automática dos conjuntos de dados, extraindo-os novamente da fonte de dados e os inserindo como uma nova versão. O **Serviço de Preservação** é responsável por arquivar conjuntos de dados. Por fim, o **Serviço de *feedback*** é responsável por gerenciar as informações de *feedback* coletadas a partir dos consumidores de dados.

[8] A = Atende à característica; AP = Atende parcialmente à característica; N = Não atende à característica

2.3. Camada de Extração

A Camada de Extração é responsável pela comunicação do Iago com os demais sistemas a partir dos quais os dados podem ser extraídos. Com a proposta de se ter um sistema o mais flexível possível, a Camada de Extração foi implementada juntamente com múltiplas bibliotecas, proporcionando o reconhecimento de várias interfaces para acesso de dados, como os diversos Sistemas Gerenciadores de Banco de Dados disponíveis no mercado.

O **Serviço de Extração de Dados** realiza a comunicação com as fontes de dados com o objetivo de extrair os dados a serem compartilhados. Uma vez extraídos, os dados são enviados para a Camada de Manipulação, em um formato abstrato, para que possa ser facilmente manipulado pelos outros serviços. Para que o Serviço de Extração possa funcionar corretamente, uma vez que o Iago se comunica com fontes distintas, se faz necessário usar **Extratores** específicos para cada tipo de fonte. Assim, cada componente Extrator inclui todo workflow de atividades de extração e transformação necessárias para extrair os dados a partir de diferentes tipos de fontes de dados.

2.4. Camada de Persistência

Por fim, a Camada de Persistência cumpre o papel de armazenar e recuperar os dados (estes já transformados e vindos da Camada de Manipulação) e metadados em um repositório, além de armazenar todas as configurações necessárias para extração dos conjuntos de dados. Esta camada também é responsável por gerenciar as diferentes versões dos conjuntos de dados.

O **Serviço de Persistência em Banco de Dados** permite manipular, organizar e acessar os dados de maneira rápida e fácil. Todos os dados e metadados referentes aos conjuntos de dados são registrados em um banco de dados. O **Serviço de Persistência em Arquivos** é responsável por armazenar versões consolidadas de conjuntos de dados de forma a reduzir as consultas diretas ao Serviço de Persistência em Banco de Dados. Por isso, os dados são transformados previamente na camada de manipulação e armazenados separadamente pelo Serviço de Persistência de Arquivos. Com essa estratégia, o sistema consegue recuperar dados com uma eficácia significante sem comprometer a eficiência.

2.5. Tecnologias Usadas

O desenvolvimento do Iago se deu a partir do uso de tecnologias que possibilitam a criação de aplicações Web. Além disso, foram escolhidas tecnologias populares e *open-source* visando manter um custo mais baixo de manutenção e adaptação do sistema, como o Java e o Angular, usados para o *back-end* e o *front-end*, respectivamente.

Em conjunto com o Java, o Spring Framework foi usado por facilitar a aplicação de padrões de desenvolvimento, ajudando a manter a qualidade do código fonte, além de ter uma comunidade ativa no desenvolvimento de novas extensões, que aumentam o alcance das funcionalidades do *framework*. No Iago, o Spring se destaca por participar da execução da maioria dos serviços oferecidos.

Além disso, O Iago faz uso de tecnologias para o armazenamento e recuperação de dados. O Apache Commons e o Hibernate são usados para recuperação de dados em arquivos de texto e Banco de Dados Relacionais, respectivamente. Já para o armazenamento dos dados a serem publicados é utilizado o MongoDB, que por ser um banco de

dados não relacional, consegue armazenar dados de forma flexível e garantir uma alta velocidade de recuperação.

3. Iago na Prática

Em nossa demonstração, exemplificamos como o Iago pode ajudar os produtores de dados a publicar conjuntos de dados. Em particular, o Iago, por meio de sua interface gráfica possibilita a configuração dos serviços de extração, publicação e atualização automática de dados. Para tanto, o produtor deverá fornecer algumas informações que serão usadas tanto para geração de metadados quanto para coordenar o processo de criação de um conjunto de dados.

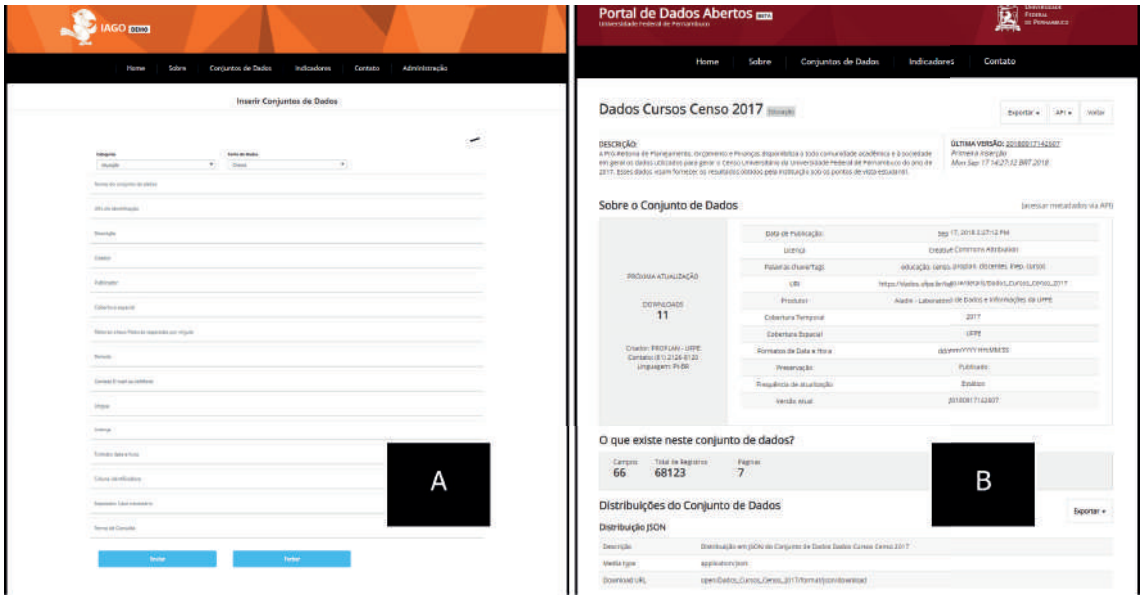


Figura 2. (A) Ilustra o momento da publicação de um conjunto de dados na versão demo. (B) Ilustra o consumo de um conjunto de dados em um portal de dados abertos que usa o Iago.

O processo para a publicação de um novo Conjunto de Dados no Iago se inicia a partir do cadastro ou seleção de um categoria já existente. Em seguida, é necessário informar ao sistema de onde os dados serão extraídos. Para isso, deve ser cadastrada uma fonte de dados. Caso a fonte de dados já tenha sido cadastrada previamente, o processo continua, caso contrário, o Iago validará a conexão com a nova fonte de dados para certificar que a extração de dados é possível.

Assim que a validação da fonte de dados é finalizada, ocorre o cadastro dos metadados que descreverão o Conjunto de Dados. O próximo passo é inserir os dados de consulta que serão utilizados como parâmetro na fonte de dados selecionada a fim de que os dados desejados sejam retornados (Ver Figura 2A). É importante destacar que os dados de consulta podem variar de acordo com o tipo de fonte de dados cadastrada. Por exemplo, se os dados forem extraídos de um banco de dados relacional, uma consulta SQL deve ser informada. Em outro cenário, no qual os dados serão extraídos de um arquivo de texto, o caminho para o arquivo deve ser parametrizado.

Assim, os metadados da primeira versão do conjunto de dados são gerados, incluindo o primeiro registro no histórico de versões e o primeiro identificador de versão.

Quando a parte cadastral é finalizada, todos os metadados que descrevem o conjunto de dados, bem como suas informações para extração na fonte de dados são finalmente armazenados no banco de dados do Iago.

A partir daqui, todo o processo é feito de maneira totalmente automática. O primeiro passo para a extração de dados é verificar se a fonte de dados está disponível no momento, caso afirmativo, o processo pode continuar, caso contrário, o sistema espera mais uma hora até que a verificação possa ser feita novamente. O tempo de uma hora representa o período em que as rotinas automáticas do Iago são executadas para verificar quais Conjuntos de Dados devem ser atualizados.

Com a verificação bem sucedida, os dados são finalmente solicitados e retornados da fonte de dados. Os dados são retornados em um formato pré-processado de maneira a facilitar o próximo passo, que transforma os dados para o formato das distribuições disponíveis no Iago e consequentemente armazenados definitivamente no sistema.

Por fim, as APIs e interfaces gráficas para acesso aos dados são finalmente criadas e disponibilizadas para o uso dos consumidores de dados. A Figura 2B apresenta um exemplo de interface gráfica para acesso a um conjunto de dados chamado ”Dados Cursos Censo 2017”. Como pode ser visto, são listados metadados descritivos, informações estruturais do conjunto de dados, e várias outras informações operacionais (e.g., quantidade de acessos, distribuições disponíveis, indicativos de versão). Todas essas informações podem ser recuperadas também através da API. No mais, esta interface também permite que supracitado conjunto de dados seja recuperado por meio de download em massa e por meio de uma API. Esse mesmo padrão de interface é fornecido para todos os conjuntos de dados armazenados pelo Iago.

A demonstração de funcionamento do Iago está disponível em: `http://tiny.cc/q6r9bz`. Um vídeo explicativo pode ser visto em: `tiny.cc/kwr9bz`.

4. Considerações Finais

Neste trabalho, apresentamos a ferramenta Iago para compartilhamento de dados na Web. O Iago leva em consideração boas práticas para publicação de dados na Web, além de procurar suprir algumas lacunas apresentada pelas soluções atuais de publicação e catalogação de dados. Como proposta de extensão para esta pesquisa, procuraremos incorporar funcionalidades para gerenciamento de séries temporais e espaciais e serviços para curadoria de dados e metadados.

Referências

Lóscio, B. F., Burle, C., and Calegari, N. (2017). Data on the Web Best Practices. W3C Recommendation, World Wide Web Consortium (W3C). Accessed in 01-July-2019.

Oliveira, L. E. R., Oliveira, M. I. S., Santos, W. C. d. R., and Lóscio, B. F. (2018). Data on the web management system: a reference model. In *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, page 2. ACM.

Oliveira, L. E. R. d. A., Oliveira, M. I. S., and Lóscio, B. F. (2017). Um survey sobre soluções para publicação de dados na web sob a perspectiva das boas práticas do w3c. In *SBBD*, pages 148–159.

SimiWork: uma Arquitetura Distribuída baseada em Workflows para Recuperação de Imagens por Conteúdo*

Gustavo Mariotto-Oliveira, Luis F. Milano-Oliveira, Daniel S. Kaster

[1]Departamento de Computação – Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – 86.057-970 – Londrina – PR – Brazil

dgustavomoliveira@gmail.com, luismilanooliveira@gmail.com, dskaster@uel.br

Resumo. *Para dados complexos, como imagens e vídeos, é comum realizar consultas por similaridade. Técnicas para Recuperação de Imagens Baseado em Conteúdo visam utilizar características visuais das imagens para realizar as consultas por similaridade. Essas consultas ocorrem após a definição de um espaço de similaridade, que é um processo complexo e depende de especialistas de domínio, que geralmente não possuem conhecimento aprofundado em computação. Este artigo apresenta o SimiWork, uma arquitetura baseada em workflows científicos permitindo que usuários possam definir espaços de similaridade encadeando tarefas e executar consultas por similaridade de forma visual. O SimiWork utiliza um ambiente distribuído com chamadas remotas para execução de algoritmos complexos e custosos computacionalmente, trazendo ao usuário alto poder computacional de forma transparente.*

Abstract. *Similarity queries are common to retrieve complex data such as images and videos. Content-based Image Retrieval techniques are intended to use visual features of images to perform similarity queries. These queries occur after defining a similarity space, which is a complex process and depends on domain experts, who usually do not have in-depth knowledge of computing. This article presents SimiWork, an architecture based on scientific workflows that allows users to define similarity spaces by chaining tasks and executing queries by similarity in an intuitive way. SimiWork uses a distributed environment with remote calls to execute complex and computationally costly algorithms, bringing the user high computational power transparently.*

1. Introdução

Recuperação de Imagens Baseado em Conteúdo (*Content-Based Image Retrieval* — CBIR) compreende um conjunto de técnicas utilizadas para recuperação de imagens a partir do seu conteúdo visual [Datta et al. 2008]. O sucesso de um sistema de CBIR normalmente demanda inúmeros passos de transformação. O objetivo é gerar uma representação em um espaço de similaridade que melhor represente o conteúdo das imagens. Como não existe uma noção universal de como definir um espaço de similaridade ideal, esse processo se torna dependente do contexto e impacta diretamente no resultado das consultas.

Existem várias técnicas que podem ser utilizadas para gerar um espaço de similaridade incluindo pré-processamento, extração de características, concatenação de vetores

*Este trabalho teve suporte financeiro da Capes e do CNPq.

de características, redução de dimensionalidade, entre outras. Qualquer mudança na ordem de execução ou dos algoritmos usados gera diferentes vetores de características e, portanto, um novo espaço de similaridade com diferentes relações de similaridade entre as imagens, produzindo em diferentes resultados de consulta [Barioni et al. 2011].

Uma estratégia para auxiliar e agilizar o processo de criação do espaço de similaridade consiste no uso de gerenciadores de *workflows* científicos (*Scientific Workflow Management Systems* — SWfMS). Um SWfMS é responsável por criar, definir, manipular e gerenciar a execução de *workflows* científicos, que são adequados para representar processos científicos complexos [Liu et al. 2015]. Além disso, SWfMS suportam atividades para a proveniência de dados, incluindo os dados originais, intermediários e as etapas computacionais de transformação envolvidas.

Este artigo apresenta o SimiWork[1], uma arquitetura com operadores para definição de espaços de similaridade através de *workflows* científicos. O objetivo do SimiWork é fornecer métodos de CBIR de maneira visual e utilizar um *cluster* Spark para execução e armazenamento dos dados. Desta forma, especialistas de diversas áreas e sem conhecimento prévio em computação podem definir diferentes espaços de similaridade de acordo com o contexto de seu trabalho e obter os resultados de análise envolvendo grandes conjuntos de imagens a partir da execução em um ambiente distribuído de forma remota.

A Seção 2 apresenta os fundamentos para o entendimento da proposta e trabalhos relacionados. A Seção 3 descreve a arquitetura *Simiwork*. A Seção 4 apresenta um estudo de caso utilizando a arquitetura e a Seção 5 a conclusão e trabalhos futuros.

2. Fundamentos e Trabalhos Relacionados

Aumentou-se o interesse em melhorar a performance das técnicas de Recuperação de Imagens Baseado em Conteúdo (CBIR) sobre grandes quantidade de dados. Datta *et al*. [Datta et al. 2008] definem CBIR como qualquer tecnologia que auxilia no gerenciamento de imagens digitais utilizando seu conteúdo visual. Em CBIR, o primeiro passo é gerar um vetor de características que represente um ou mais padrões visuais como cor, textura ou forma. A similaridade entre duas imagens é inversamente proporcional à distância entre os vetores de características que as representam. Assim sendo, uma instância do espaço de similaridade é definida pelo vetor de características e a função de distância, e qualquer alteração realizada a um destes elementos produz uma nova instância [Barioni et al. 2011]. Adicionalmente, existem outras técnicas que podem ser utilizadas, como seleção de características ou processamento das imagens. O objetivo é produzir uma instância do espaço de similaridade que melhor represente a distribuição de similaridade entre as imagens de um conjunto.

O processo de definição do espaço de similaridade é complexo, caracterizado por dois fatores principais. O primeiro fator é o ponto de vista semântico, ou seja, as características importantes para cada domínio de imagens está diretamente ligado à um especialista. Porém é comum que este especialista não tenha conhecimento prévio em computação, e este é o segundo fator que promove a complexidade no processo [Oliveira and Kaster 2017]. Para contornar essa situação os gerenciadores de *workflows* científicos (SWfMS) oferecem ao usuário as tarefas que podem ser executadas,

[1]Link para o vídeo de demonstração: `http://www.uel.br/cce/dc/?p=2047`

este, por sua vez, define uma sequência de execução na forma de grafos acíclicos [Liu et al. 2015], possivelmente removendo a barreira do especialista sem conhecimento amplo em computação.

Existem vários sistemas de CBIR na literatura. Entretanto, o número de propostas que utilizam processamento distribuído e sistemas de *workflow* é bastante reduzido. Um exemplo é o projeto *VISCERAL* [Hanbury et al. 2013], que tem como objetivo oferecer um ambiente onde pesquisadores desenvolvem soluções para problemas previamente definidos utilizando dados públicos, e assim, comparar seus resultados com resultados de outras soluções. Porém, o volume de dados para realização das tarefas tornou-se grande, sendo necessário a utilização de ambiente distribuído. Além disso, frequentes alterações nos *datasets* dificultou a comparação entre os algoritmos. Diferente da proposta do projeto *VISCERAL*, a arquitetura descrita neste trabalho tem como alvo usuários que não possuem o conhecimento em computação, mas que desejam utilizar das técnicas de CBIR para sua área de domínio. Seguindo essa linha, Valente *et al*. [Valente et al. 2014] propuseram um protótipo baseado em *workflows*, definidos através de linhas de comandos para execução de algortimos de aprendizado de máquina com uma infraestrutura em grade para análise de imagens médicas em grande escala. Outro trabalho também propõe a construção de *pipelines* de execução utilizando linhas de comandos, e permite a visualização de resultados intermediários, os estados das tarefas e o grafo gerado pela *pipeline*, através de uma interface gráfica [Sridharan 2015]. Este trabalho, em contrapartida, diferencia-se no uso de tecnologias livres e fácil configuração e na criação de *pipelines* de execução através de uma interface gráfica e o uso de um *framework* baseado no paradigma *map-reduce* bastante difundido atualmente.

3. O SimiWork

A Figura 1 apresenta os componentes da arquitetura SimiWork e a integração entre os diferentes serviços. O usuário interage com a arquitetura a partir do SWfMS Taverna Workbench[2] e de um navegador *web*. O navegador *web* acessa o Gerenciador Web de Arquivos do Simiwork, que permite gerenciar conjuntos de imagens e acessar resultados de processamento, que também podem ser conjuntos de dados (por exemplo, um conjunto de vetores de características). O usuário pode utilizar um conjunto de dados existente no sistema de arquivos HDFS da arquitetura ou enviar um novo, via *upload* ou carga a partir de um serviço de armazenamento em nuvem. Uma vez que um conjunto de dados foi enviado para o HDFS o usuário pode utilizá-lo de qualquer lugar, já que esses dados estão armazenados em um *cluster* e o acesso é remoto através de um navegador *web*. No Taverna, o usuário define e executa workflows. As atividades oferecidas pelo SimiWork incluem algoritmos de pré-processamento de imagens (*e.g.*, aumento de constraste e redução de ruído), extratores de características de imagens (*e.g.*, Haralick e AutoColorCorrelogram), algoritmos de redução de dimensionalidade (*e.g.*, PCA), operações de consulta por similaridade (*e.g.*, consulta aos k-vizinhos mais próximos) e métodos de análise de resultados (*e.g.*, dados de precisão e revocação).

A execução das atividades do *workflow* consiste em invocações de serviços *web*, gerenciadas pelo servidor de serviços *web* Apache Axis2. O Axis2 fornece uma interface baseada na linguagem de descrição de serviços WSDL (*Web Services Description Lan-*

[2]https://taverna.incubator.apache.org/

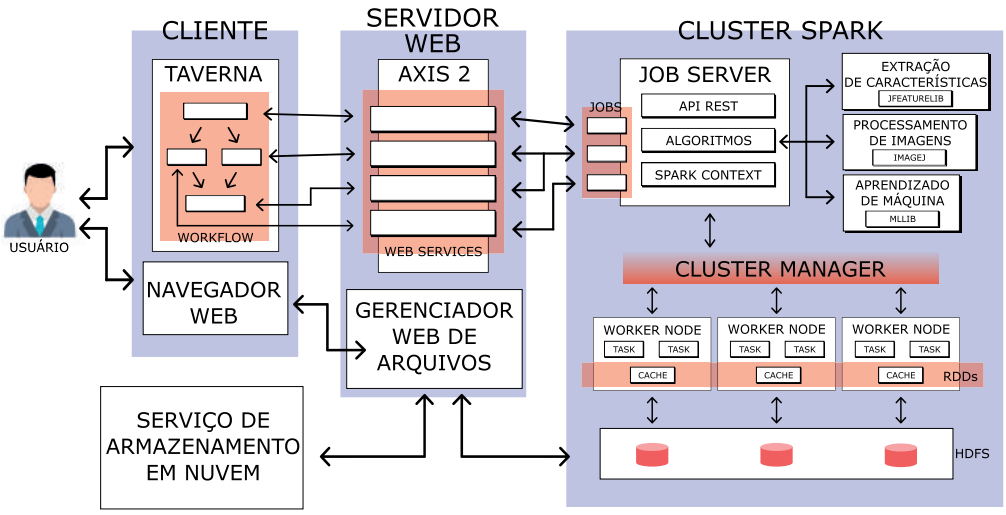


Figura 1. Componentes da arquitetura SimiWork.

guage), que também é suportada pelo Taverna e permite estabelecer a ligação entre as atividades dos *workflows* e os serviços *web*. Os serviços *web* enviam requisições para o Spark JobServer, que é um sistema que permite a execução de *jobs* no *framework* Apache Spark com controle de contextos. O JobServer permite que vários *jobs* compartilhem o mesmo contexto, o que é uma estratégia fundamental do SimiWork para permitir a execução de atidades do *workflow* conectando entradas e saídas diretamente no *cluster*. O Taverna apenas orquestra a execução, então não há transferências significativas de dados entre o cluster e o cliente. Através do Taverna também é possível armazenar os dados utilizados como entrada e as saídas de cada atividade de um *workflow*. Esses dados podem ser usados posteriormente para recuperação ou reutilização de resultados intermediários.

O Spark é o responsável por controlar os recursos do cluster e garantir tolerância à falhas em nível de aplicação. Ele recebe do JobServer toda informação que precisa para processar os dados conforme as especificações do usuário, incluindo os parâmetros e o conjunto de dados de entrada para os algoritmos, que podem ser saídas produzidas por algoritmos de atividades anteriores. Na versão atual do SimiWork, os algoritmos de processamento de imagens são da biblioteca ImageJ[3], os de extração de características são da biblioteca JFeatureLib[4] e os de redução de dimensionalidade são da MLLib[5].

Os conjuntos de dados, conjuntos de imagens de entrada e conjuntos de dados salvos pelo SimiWork, são armazenados nos nós da arquitetura, com uso do sistema de arquivos distribuído HDFS (*Hadoop Distributed File System*), que oferece tolerância a falhas. Contudo, o Spark utiliza uma abstração conhecida como RDD (*Resilient Distributed Dataset* — conjuntos de dados distribuídos e resilientes), que são coleções particionadas entre os nós de *cluster* e que podem ser operadas em paralelo. Os RDDs, uma vez criados, são mantidos em memória, para diminuir o tempo de acesso a dados, mas o SimiWork também permite persistir no HDFS conjuntos de dados em RDDs, para que o usuário tenha acesso a conjuntos dados produzidos para uso externamente à arquitetura.

[3]https://imagej.nih.gov/ij/
[4]https://github.com/locked-fg/JFeatureLib
[5]https://spark.apache.org/mllib/

A implementação contém ainda códigos para integrar todos os componentes e funções auxiliares, por exemplo, operadores de consulta por similaridade, funções de distância e abstrações para os vetores de características.

4. Estudo de Caso

Esta seção apresenta a perspectiva do usuário na utilização do SimiWork. O usuário final realiza a interação somente com o Taverna e com o *Gerenciador Web de Arquivos*. Inicialmente, o usuário cria ou seleciona um conjunto de dados e utiliza o caminho (*path*) para o conjunto na construção do *workflow*. Neste estudo de caso, foi utilizado um conjunto de imagens médicas no formato DICOM[6] e um processo ilustrativo de extração de características e análise de resultados. O processo contém duas partes e o *workflow correspondente* é apresentado na cópia de tela do Taverna na Figura 2. A primeira é a etapa de extração de características, que compreende abrir as imagens (*OpenImage*), extrair características utilizando dois extratores (*AutoColorCorrelogram* e *Haralick*), concatenar os vetores de características (*FeatureConcat*) e fazer uma redução de dimensionalidade (*PCA)*. A segunda parte é a análise de resultados por meio da geração de dados de precisão e revocação, que executa um conjunto de consultas aos k-vizinhos mais próximos (*kNNQuery*), efetua o cálculo de precisão e revocação a partir dos resultados das consultas (*Precision_Recall*) e grava o RDD final, com os dados de precisão e revocação, em um arquivo texto no HDFS (*RDDToTextFile*).

No Taverna, o usuário acessa os serviços *web* disponíveis no SimiWork importando as URLs dos arquivos WSDL fornecidos (parte de cima do menu na Figura 2). Após a importação dos serviços, eles ficam disponíveis na parte de baixo do menu e o usuário pode adicioná-los como atividades ao *workflow* arrastando-os para a área do *workflow* e conectando-os usando portas de entrada e saída. A conexão entre atividades utiliza um recurso do Taverna denominado *XML Splitters*, que faz a codificação e decodificação de parâmetros e respostas em XML para a comunicação com os serviços *web*.

Para cada tarefa, o Taverna realiza a chamada para o serviço *web* através do Axis2, passando os parâmetros da tarefa. A chamada é realizada se, e somente se, os dados de entradas estiverem prontos para serem consumidos. O serviço *web* faz uma requisição POST para o JobServer, enviando um arquivo JSON com os parâmetros. O JobServer verifica os parâmetros recebidos, recupera o RDD relativo às imagens e invoca a execução da tarefa no Spark. O gerenciador do *cluster* realiza a execução da tarefa utilizando os nós do *cluster* e retorna ao JobServer o RDD resultante da tarefa, que é salvo dentro do contexto. O JobServer retorna um arquivo JSON contendo o nome referente ao RDD de saída ao serviço *web* que, por sua vez, devolve ao Taverna um documento XML contendo este nome para ser utilizado na tarefa subsequente.

5. Conclusão

Este artigo apresentou o SimiWork, uma arquitetura com execução distribuída para suportar o processo de recuperação de imagens por conteúdo por meio de *workflows* científicos. Através do SimiWork, usuários podem executar algoritmos envolvidos em tarefas de CBIR em um ambiente de alto desempenho de forma visual através de chamadas remotas a um cluster de processamento. Desta forma, o SimiWork promove um avanço na

[6]https://data.mendeley.com/datasets/rscbjbr9sj/2

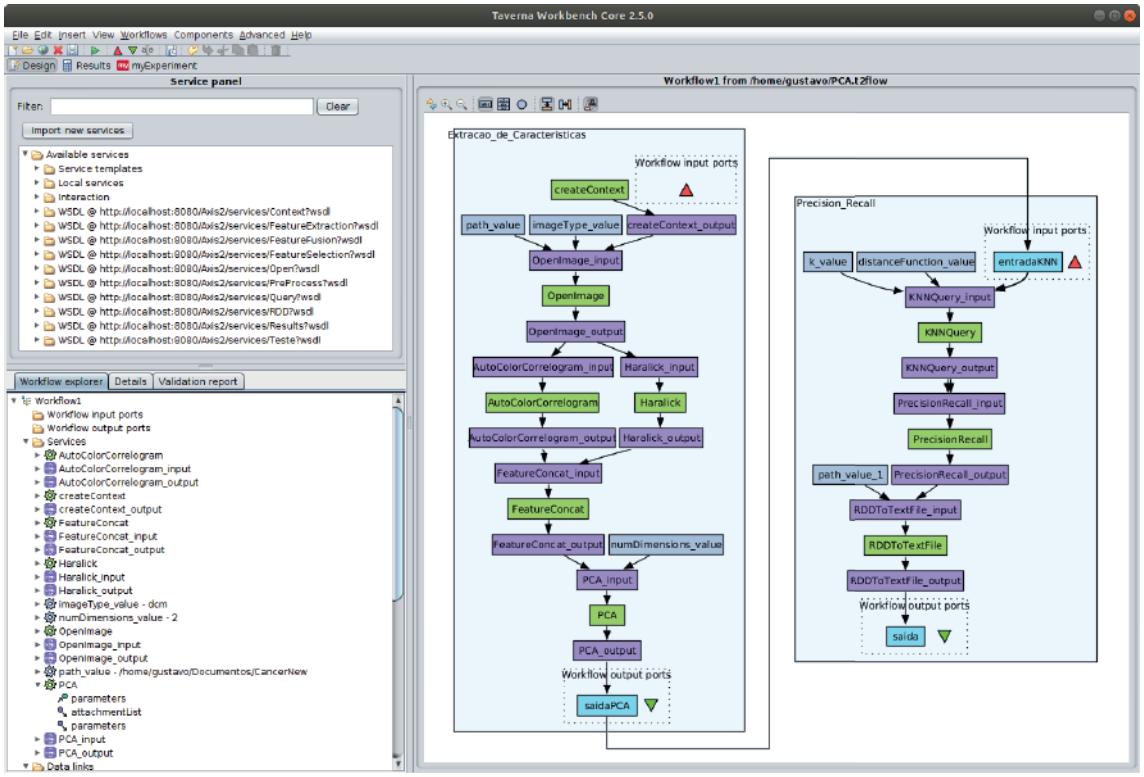


Figura 2. Tela do Taverna Workbench com o *workflow* do estudo de caso.

utilização de técnicas complexas de maneira mais intuitiva. Dentre os trabalhos em andamento e futuros destacam-se o desenvolvimento de um módulo de execução de consultas por similidade utilizando índices para dados complexos e a inclusão de mais serviços à arquitetura.

Referências

Barioni, M., Kaster, D., Razente, H., Traina, A., and Traina Jr, C. (2011). Advanced database query systems: Techniques, applications and technologies.

Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5.

Hanbury, A., Müller, H., Langs, G., and Menze, B. H. (2013). Cloud–based evaluation framework for big data. In *The Future Internet Assembly*, pages 104–114. Springer.

Liu, J., Pacitti, E., Valduriez, P., and Mattoso, M. (2015). A survey of data-intensive scientific workflow management. *Journal of Grid Computing*, 13(4):457–493.

Oliveira, L. F. M. and Kaster, D. d. S. (2017). Defining similarity spaces for large-scale image retrieval through scientific workflows. In *Proceedings of the 21st International Database Engineering & Applications Symposium*, pages 57–65. ACM.

Sridharan, R. (2015). *Visualization and analysis of large medical image collections using pipelines*. PhD thesis, Massachusetts Institute of Technology.

Valente, F., Silva, A., Costa, C., Franco, J. M., Valiente, C. S.-O., and Guevara, M. (2014). A dataflow-based approach to the design and distribution of medical image analytics. In *8th Iberian Grid Infrastructure Conference Proceedings*, page 201.

PhenoManager: um Sistema de Gerência de Hipóteses de Fenômenos Científicos*

Leonardo Ramos[1], Kary Ocaña[2], Douglas de Oliveira[2], Fabio Porto[2], Daniel de Oliveira[1]

[1]Instituto de Computação - Universidade Federal Fluminense (IC/UFF) - Brasil

leoslramos@id.uff.br, danielcmo@ic.uff.br

[2]Laboratório Nacional de Computação Científica (LNCC) - Brasil

{karyann,ericsonmarc,fporto}@lncc.br

Resumo. *Experimentos científicos baseados em simulações computacionais envolvem a gerência de grande volume de dados e metadados que são produzidos durante o ciclo de vida de um experimento científico. Tal ciclo se inicia com a identificação de um fenômeno, a formulação de uma hipótese, a modelagem e execução da simulação associada ao fenômeno até sua avaliação. A avaliação de uma hipótese pode requerer a execução de diversos experimentos diferentes, que por sua vez, podem demandar execuções de diferentes Workflows científicos ou scripts complexos, o que torna a tarefa bastante árdua, pois os mesmos podem ser executados em diferentes Sistemas de Gerência de Workflows (SGWf) e ambientes distribuídos. Atualmente, cada SGWf ou script gerencia uma simulação de forma isolada, não permitindo analisar resultados dessas simulações associadas ao mesmo experimento de forma integrada. Este artigo apresenta uma abordagem chamada PhenoManager que tem como objetivo auxiliar os cientistas a gerenciar os fenômenos observados e as hipóteses definidas em conjunto com os resultados das simulações computacionais (que podem executar em múltiplos sistemas e ambientes). O PhenoManager é capaz de auxiliar o cientista na estruturação, validação e reprodução de hipóteses de um fenômeno, por meio de modelos computacionais configuráveis, além de prover uma API de consulta e exportação de metadados por meio de Research Objects.*

1. Introdução

Nos últimos anos, houve um crescimento na utilização de simulações computacionais em experimentos científicos [Hey et al. 2012]. De acordo com [Hey et al. 2012], os experimentos científicos de hoje se baseiam fortemente na análise de dados gerados a partir de complexas simulações computacionais. Existem diversas abordagens existentes para modelar, gerenciar, monitorar e depurar experimentos baseados em simulações. Muitos usuários implementam seus próprios *scripts* e programas, enquanto que outros modelam seus experimentos utilizando Sistemas de Gerência de *Workflows* (SGWf) [de Oliveira et al. 2019], *Gateways* científicos [Ocaña et al.] ou *frameworks* MapReduce como o Apache Spark ou o Hadoop [Karau et al. 2015].

Entretanto, nenhuma dessas abordagens é capaz de representar todos os conceitos envolvidos no método científico [Mattoso et al. 2008]. Atualmente, as abordagens existentes focam somente em representar as simulações que são executadas no contexto de um determinado experimento [Deelman et al. 2009]. Porém, o ponto de partida de uma investigação científica é a descrição de um fenômeno, seja ele natural ou não. O fenômeno estudado ocorre em algum espaco-tempo, em que se observam quantidades físicas selecionadas [Porto et al. 2015]. As hipóteses científicas interpretam conceitualmente o fenômeno estudado por meio de sua representação e através de modelos matemáticos. O teste de hipóteses *in silico* envolve a execução de experimentos, representando os modelos matemáticos e confrontando dados gerados a partir de simulações computacionais complexas [Porto et al. 2015]. Ou seja, de forma a confirmar ou refutar uma hipótese,

PhenoManager video: https://www.facebook.com/uffescience/

experimentos devem ser definidos, e esses experimentos podem demandar a execução de diversas simulações implementadas de diferentes formas, conforme apresentado na Figura 1 (um *workflow* no SGWf Pegasus, um *script* Python e uma aplicação MapReduce implementada no Apache Spark).

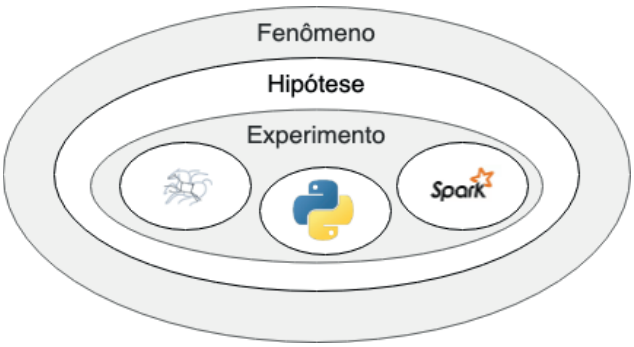


Figura 1. Relação entre os conceitos do método científico

Dessa forma, como não há uma associação entre a execução das simulações, os fenômenos observados e as hipóteses definidas, fica a cargo do cientista gerenciar todo esse conhecimento de forma manual e *ad-hoc* (propensa à erros). Ou seja, se o cientista necessitar descobrir quais execuções de um determinado *script* ou *workflow* foram essenciais para refutar a hipótese α, ele terá que registrar essas informações por conta própria. Em um cenário ainda mais complexo, a validação de uma hipótese científica pode demandar a execução de diversos *workflows*, *scripts* ou aplicações MapReduce distintas que podem executar em ambientes distribuídos e de alto desempenho, como as nuvens de computadores e supecomputadores.

Assim, seria interessante que os cientistas tivessem acesso à uma abordagem que auxiliasse na gerência do projeto científico como um todo e no apoio ao método científico, ajudando na documentação, compartilhamento dos dados obtidos e na facilitação da reprodução dos experimentos realizados, o que representa um desafio em aberto. Sendo assim, este artigo apresenta uma abordagem chamada $PhenoManager$ que visa apoiar a gerência e validação de hipóteses científicas de forma integrada à execução dos experimentos, seja via *workflows*, *scripts* ou aplicações MapReduce. O $PhenoManager$ aborda desde a etapa de concepção, de configuração do modelo de execução, até a validação e reprodução dos experimentos, por meio dos dados de proveniência [Freire et al. 2008]. Além disso, o $PhenoManager$ permite que o cientista execute experimentos em ambientes de computação de alto desempenho (como o supercomputador Santos Dumont[1] do Laboratório Nacional de Computação Científica - LNCC) e se integra com o sistema *SciManager* [Ramos et al. 2016], que gerencia tarefas de equipes em projetos científicos, em um mesmo ecossistema de *software*.

O restante do artigo está organizado em três seções além da introdução. A Seção 2 apresenta uma visão geral do $PhenoManager$. A Seção 3 discute como será realizada a demonstração do $PhenoManager$ e, finalmente, a Seção 4 conclui o presente artigo.

2. Arquitetura do $PhenoManager$

A Figura 2 apresenta a arquitetura do $PhenoManager$ e seus componentes principais. O $PhenoManager$ pode ser dividido em seis camadas funcionais: (i) Camada de Autenticação, (ii) Camada de Gerência do Ambiente, (iii) Camada de Execução, (iv) Camada de Dados, (v) Camada de Consulta, e (vi) Portal *Web*. A seguir discutimos em detalhes cada uma dessas camadas.

A Camada de Autenticação é a responsável por gerenciar as credenciais de acesso ao $PhenoManager$. Essa camada é fundamental, uma vez que dados de pesquisas não publicados

[1] `https://sdumont.lncc.br/`

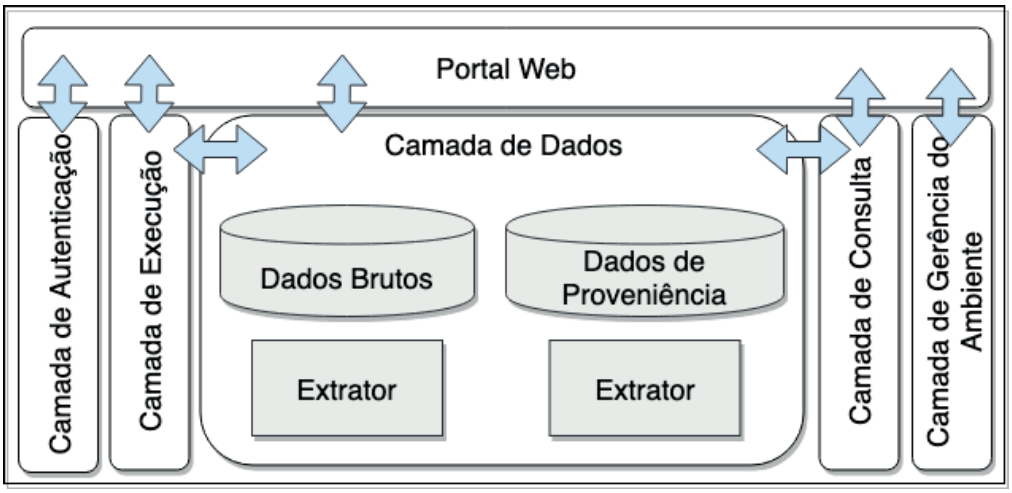


Figura 2. Arquitetura do $PhenoManager$

serão manipulados pelo sistema. O controle de credenciais é realizado em dois níveis. No nível do "Perfil Pessoal", cada usuário da ferramenta configura/preenche suas informações pessoais e carrega credenciais para acesso aos diversos ambientes (*e.g.*, Amazon AWS). No nível de "Grupos de Usuários", o(s) usuário(s) administrador(es) (com privilégios para criar grupos) procuram, selecionam e agrupam perfis de usuários existentes, e, a partir da criação do grupo, compartilham os mesmos privilégios no ambiente. Cada usuário ou grupo poderá ter acesso aos dados e funcionalidades providos pelo sistema de acordo com os seguintes privilégios: (i) Permissão para leitura ("*READ*"): o usuário e/ou grupo pode apenas visualizar os dados, porém não pode editar e/ou cadastrar qualquer dado; (ii) Permissão para escrita ("*WRITE*"): o usuário e/ou grupo pode ler e cadastrar informações dentro do contexto especificado, tornando-o membro ativo do sistema; (iii) Permissão de administrador ("*ADMIN*"): além das permissões anteriormente citadas, o usuário e/ou grupo pode cadastrar permissões para outros usuários e grupos;

A Camada de Gerência do Ambiente é a responsável por configurar ambientes distribuídos para a execução das simulações. Para cada ambiente diferente, um componente deve ser desenvolvido com as chamadas para a API específica do mesmo. O $PhenoManager$ já provê nativamente a integração com três tipos de ambientes diferentes: *Cluster*, *Cloud (Amazon AWS)* e *SSH*. Além disso, é possível configurar uma conexão *VPN* para estes ambientes, podendo selecionar entre *Cisco VPN* e *VPN default*. Para o ambiente *Cloud*, é possível configurar no detalhe, os tipos e imagens das máquinas virtuais que serão construídas no ambiente da *Amazon AWS*.

A Camada de Execução é a responsável por invocar SGWfs, *scripts* ou aplicações externas nos ambientes de alto desempenho. Para cada sistema diferente ou aplicação a ser invocada, um *wrapper* específico deve ser provido (já que o $PhenoManager$ necessita conhecer o processo de invocação da aplicação externa). A chamada aos *wrappers* é assíncrona, logo o serviço dessa camada pode ser escalado em mais instâncias, aumentando, dessa forma, o *throughput* de execuções paralelas de simulações científicos para diferentes usuários. Sendo assim, por meio desse artifício, procuramos garantir o paralelismo e a alta disponibilidade do $PhenoManager$.

A Camada de Dados contém o banco de dados de proveniência (com todos os metadados registrados no $PhenoManager$) e os dados brutos produzidos pelas simulações computacionais. Além disso, essa camada contém uma série de extratores responsáveis por acessar a base de proveniência ou o *log* da aplicação externa e carregar as informações no banco de dados de proveniência do $PhenoManager$. Em sua versão atual, o banco de dados de proveniência se encontra modelado no PostgreSQL e os dados brutos são carregados no Google Drive.

A Camada de Consulta é a responsável por permitir que o cientista possa submeter con-

sultas aos dados gerenciados pelo $PhenoManager$. Essa camada provê uma API que abstrai as consultas aos dados de modo a facilitar sua manipulação por cientistas e outras aplicações consumidoras deste serviço. A API permite que o cientista submeta consultas contendo filtros, ordenações, funções de agregação e projeções de campos em todas as entidades expostas no modelo de dados do $PhenoManager$. Outro ponto importante é que a *API* só responde com sucesso se as credenciais corretas forem passadas no cabeçalho da solicitação. Além disso, a Camada de Consulta é responsável por exportar pacotes chamados *Research Objects* (RO) [Holl et al. 2013], que contém tanto os metadados consultados quanto os dados brutos produzidos pela simulação. Por meio dos ROs, os cientistas são capazes de reproduzir uma determinada simulação, um experimento ou verificar se uma hipótese foi efetivamente validada.

Finalmente, o Portal *Web* é o responsável por toda a interface com o cientista e a integração com as demais camadas. Nele o cientista registra os fenômenos observados, as hipóteses associadas, seus experimentos e os modelos que executam as simulações de cada experimento (*e.g., workflow*, *script* ou aplicação). Além disso, por meio do Portal *Web*, o cientista é capaz de executar efetivamente suas simulações e consultar os dados de proveniência coletados de forma integrada, *i.e.,* se um mesmo experimento for composto de diversos *workflows* e aplicações, as consultas à base de proveniência consideram todas as simulações como parte do mesmo experimento, o que não ocorre nas ferramentas existentes que gerenciam os modelos computacionais de maneira isolada [Deelman et al. 2009].

Em termos de implementação, o $PhenoManager$ foi desenvolvido na linguagem Java e segue o padrão arquitetural de APIs como microserviços, ou seja, cada componente é um serviço *Web* autônomo e pequeno que disponibiliza apenas uma funcionalidade [Newman 2015]. Todos os microserviços foram construídos por meio do *framework Spring Boot*, que já oferece apoio para desenvolvimento de aplicações nesse padrão de uma maneira rápida e pouco verbosa. Para segurança de dados e autenticação entre os componentes, foi utilizado o arcabouço *Spring Security*. O desenvolvimento das interfaces, templates e telas do Portal *Web* foi realizado com *AngularJs*. Como cada serviço que compõe a arquitetura do $PhenoManager$ é completamente isolado dos demais, a escalabilidade se torna um dos pontos chave deste ecossistema. Para garantir a invocação de aplicações externas de forma assíncrona, foi escolhido o *message Broker* de código aberto RabbitMQ.

3. Demonstração do $PhenoManager$

Propõe-se aos participantes do SBBD um cenário de uso para que os mesmos tenham uma visão completa do $PhenoManager$ (Figura 3). A demonstração se inicia com o *dashboard* do $PhenoManager$ (Figura 3(a)) que apresenta todas as execuções de simulações em andamento, finalizadas, *etc* para controle do cientista. Após, é necessário o cadastro do fenômeno e hipóteses no $PhenoManager$. Basicamente, o usuário deve informar um nome e uma descrição tanto para o fenômeno quanto para a hipótese. Após, os usuários devem registrar os seus *scripts* ou *workflows* no $PhenoManager$ (Figura 3(b)). É importante ressaltar que os modelos que executam as simulações são chamados de *Executors* no $PhenoManager$. Além dos modelos, o cientista deve configurar o ambiente de execução (Figura 3(c)). Uma vez que tanto os modelos quanto os ambientes estão configurados, o usuário deve carregar os dados no $PhenoManager$ por meio da interface *Web*. Com os dados carregados, os modelos podem ser executados e o usuário pode monitorar as execuções dos mesmos (Figura 3(d)). Serão fornecidos dados de exemplo, mas os usuários são encorajados a realizar a carga de seus próprios dados e *scripts*.

Uma vez que as execuções se iniciem ou já tenham terminado, o usuário é capaz de consultar a base de proveniência do $PhenoManager$. Um exemplo de consulta que pode ser submetida é ”Quais os nomes e versões dos modelos utilizados na validação de uma hipótese com um nome específico (“sciphy”)?”. Essa consulta é invocada por meio da chamada à API de consulta do $PhenoManager$ apresentada na Figura 4. Após o processamento da consulta é gerado um *Re-*

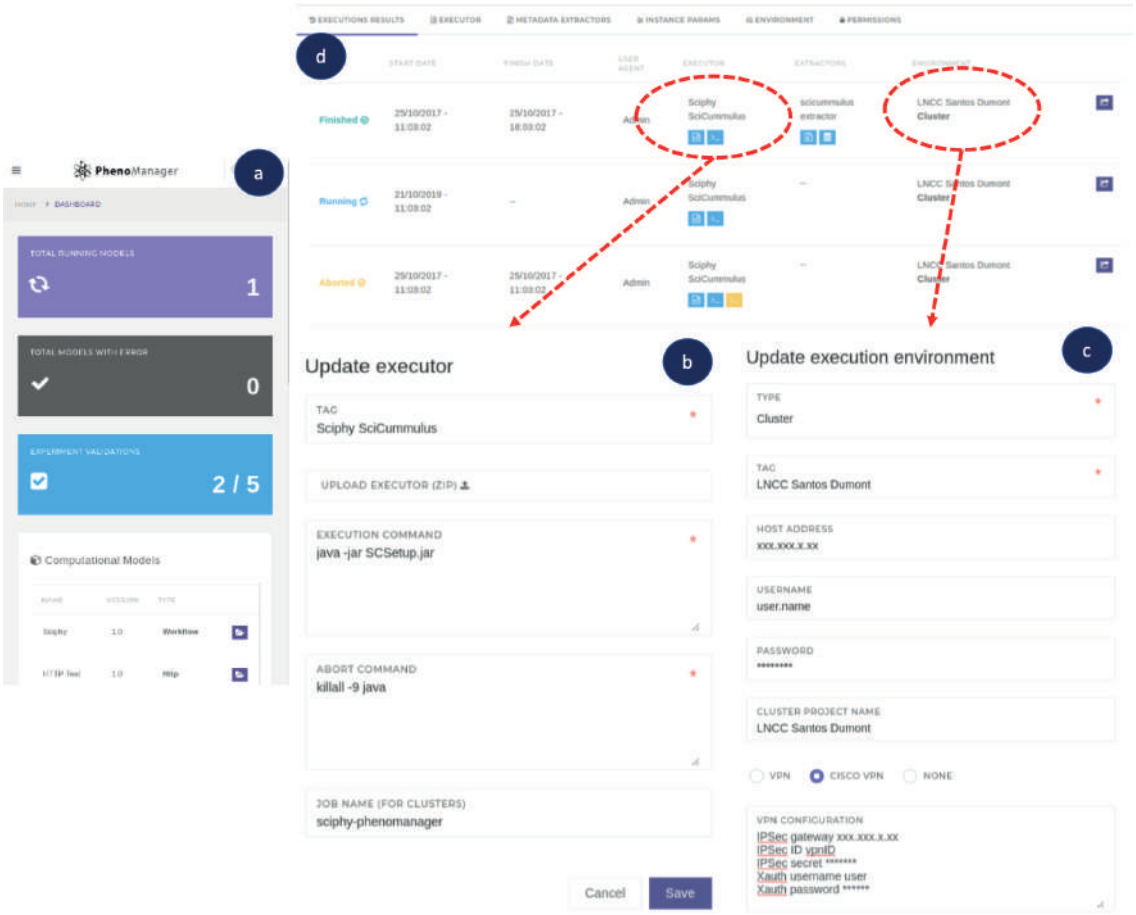


Figura 3. Interface do $PhenoManager$

search Object contendo a resposta da consulta e os dados associados para *download*. A Figura 5 apresenta um fragmento do *Research Object* gerado.

```
http://phenomanager.ic.uff.br/PhenoManagerApi/v1/computational_models?projectio
n=[name,currentVersion]&filter=[experiment.hypothesis.name=like=sciphy;experime
nt.hypothesis.state=VALIDATED]
```

Figura 4. Exemplo de Consulta na API do $PhenoManager$

4. Conclusões

O presente artigo apresenta o $PhenoManager$, um Sistema de Gerência de Hipóteses de Fenômenos científicos que é capaz de gerenciar fenômenos e hipóteses em conjunto com a execução dos seus experimentos e simulações computacionais associadas. O $PhenoManager$ se baseia em uma arquitetura de microserviços, o que facilita a sua extensão e escalabilidade. Dessa forma, o $PhenoManager$ fornece um valioso ponto de partida para a análise integrada de dados de proveniência de múltiplos sistemas e programas. Como trabalho futuro, pretendemos fornecer análises adicionais como por exemplo, análises de desempenho das simulações executadas. Além disso, pretendemos implantar um mecanismo automático de validação das hipóteses a partir dos dados de proveniência coletados. O $PhenoManager$ pode ser obtido em `https://github.com/UFFeScience/Phenomanager`.

Agradecimentos

A pesquisa apresentada neste artigo foi parcialmente financiada por CNPq, CAPES e FAPERJ.

Figura 5. Fragmento do *Research Object* gerado pelo $PhenoManager$

Referências

de Oliveira, D. C. M., Liu, J., and Pacitti, E. (2019). *Data-Intensive Workflow Management: For Clouds and Data-Intensive and Scalable Computing Environments*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

Deelman, E., Gannon, D., Shields, M., and Taylor, I. (2009). Workflows and e-science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5):528–540.

Freire, J., Koop, D., Santos, E., and Silva, C. T. (2008). Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3).

Hey, T., Gannon, D., and Pinkelman, J. (2012). The future of data-intensive science. *IEEE Computer*, 45(5):81–82.

Holl, S., Garijo, D., Belhajjame, K., Zimmermann, O., Giovanni, R. D., Obst, M., and Goble, C. A. (2013). On specifying and sharing scientific workflow optimization results using research objects. In *WORKS 2013, Denver, CO, USA, November 17, 2013*, pages 28–37.

Karau, H., Konwinski, A., Wendell, P., and Zaharia, M. (2015). *Learning spark: lightning-fast big data analysis*. "O'Reilly Media, Inc.".

Mattoso, M., Werner, C., Travassos, G., Braganholo, V., and Murta, L. (2008). Gerenciando experimentos científicos em larga escala. *SBC-SEMISH*, 8:121–135.

Newman, S. (2015). *Building microservices: designing fine-grained systems*. "O'Reilly Media, Inc.".

Ocaña, K., Galheigo, M., Osthoff, C., Jr., L. M. R. G., Gomes, A. T. A., de Oliveira, D., Porto, F., and de Vasconcelos, A. T. R. Towards a science gateway for bioinformatics: Experiences in the brazilian system of high performance computing. In *CCGRID*.

Porto, F., Costa, R. G., de Carvalho Moura, A. M., and Gonçalves, B. (2015). Modeling and implementing scientific hypothesis. *J. Database Manag.*, 26(2):1–13.

Ramos, L. S., Ocaña, K. A., and de Oliveira, D. (2016). Um sistema de informação para gerência de projetos científicos baseados em simulações computacionais. In *Anais do XII Simpósio Brasileiro de Sistemas de Informação*, pages 216–223. SBC.

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Workshop on Thesis and Dissertations in Databases

PROCEEDINGS

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

Program Chair

Jonice Oliveira (UFRJ, Brazil)

Editorial

The Thesis and Dissertation Workshop (WTDBD – Workshop de Teses e Dissertações em Banco de Dados), an event of the Brazilian Symposium on Databases (SBBD), is a forum for discussing ongoing Brazilian graduate work on databases. The main goal is to present proposals and approaches that are currently under development (as opposed to concluded work that is presented in SBBD). During the workshop, each paper is presented to a committee of three researchers of related areas. The committee then provides a critical analysis of the work and suggests possible issues to be worked on. Such a high-quality feedback may be extremely valuable for graduate students who are in the initial or middle phase of their work.

This year, WTDBD had 27 submissions, from which 21 were selected for presentation. All papers received at least 3 reviews.

We would like to thank the members of the program committee and external reviewers, who made excellent reviews with positive feedback to all papers. These reviews will certainly improve the quality of the final work. Our gratitude also goes to the local organizers, SBBD 2019 chairs and steering committee, who helped us to accommodate WTDBD sessions within the full program.

Finally, special thanks to students and their advisors who submitted papers to WTDBD. This event does not exist without you.

Have all a wonderful event!

Jonice Oliveira (UFRJ, Brazil)
Thesis and Dissertation Workshop Chair

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

SBBD Steering Commitee

Ângelo Brayner (UFC)
Bernadette Lóscio (UFPE) coordenadora da CEBD
Carina Dorneles (UFSC)
Sérgio Lifschitz (PUC-Rio)
Fábio Porto (LNCC)
Carmem Hara (UFPR)

SBBD 2019 Commitee

Steering Committee Chair
Bernadette Lóscio (UFPE)

Local Chair:
José Maria da Silva Monteiro Filho (UFC, Brazil)

Full Paper Chair
Carina F. Dorneles (UFSC, Brazil)

Short Paper Chair
Fábio Porto (LNCC, Brazil)

Demos and Applications Chair
Robson L. F. Cordeiro (ICMC-USP, Brazil)

Thesis and Dissertation Workshop Chair
Jonice Oliveira (UFRJ, Brazil)

Tutorials Chair
Altigran Soares da Silva (UFAM, Brazil)

Short course Chair
Maria Cláudia Cavalcanti (IME, Brazil)

Workshop Chair
José Antônio Macedo (UFC, Brazil)

Thesis and Dissertation Contest Chair
Caetano Traina Jr. (USP, Brazil)

Graduation Student Workshop Chair
Ticiana Linhares (UFC, Brazil)

Local Organization Committee
SBBD Local Chair: José Maria da Silva Monteiro Filho (DC/UFC)
Leonardo Oliveira Moreira (Instituto UFC Virtual/UFC)
Marum Simão Filho (UNI7)
Angelo Roncalli de Alencar Brayner (DC/UFC)
Javam de Castro Machado (DC/UFC)

Thesis and Dissertation Workshop Program Committee
Alexandre Plastino (Universidade Federal Fluminense)
Altigran Soares da Silva (Universidade Federal do Amazonas)
Ana Carolina Almeida (Universidade do Estado do Rio de Janeiro)
Anderson Ferreira (Universidade Federal de Ouro Preto)
Angelo Brayner (Federal University of Ceara - UFC)
Bernadette Loscio (Universidade Federal de Pernambuco)
Carina F. Dorneles (Universidade Federal de Santa Catarina)
Carlos Eduardo Barbosa (COPPE/UFRJ)
Carlos Eduardo Santos Pires (Federal University of Campina Grande)
Carmem Hara (Universidade Federal do Paraná)
Clodoveu Davis (UFMG)
Cristina Ciferri (USP)
Daniel de Oliveira (Universidade Federal Fluminense)
Daniel Kaster (UEL - Universidade Estadual de Londrina)
Denio Duarte (UFFS)
Dimas C. Nascimento (Universidade Federal Rural de Pernambuco - UFRPE)
Duncan Ruiz (Pontifícia Universidade Católica do RS)
Edleno Moura (Federal University of Amazonas)
Eduardo de Almeida (Universidade Federal do Paraná)
Eduardo Ogasawara (CEFET/RJ)
Elaine Sousa (University of Sao Paulo)
Fabio Porto (LNCC)
Flavio Horita (Universidade Federal do ABC)
Giseli Lopes (UFRJ)
Humberto Razente (Universidade Federal de Uberlandia)
João Eduardo Ferreira (IME/USP)
José de Aguiar Moraes Filho (University of Fortaleza - UNIFOR)
Karin Becker (UFRGS)
Kary Ocaña (LNCC)

Kelly Braghetto (IME/USP)
Lívia Ruback (Universidade Federal do Rio de Janeiro)
Leonardo Moreira (Universidade Federal do Ceará)
Marcos Gonçalves (Universidade Federal de Minas Gerais)
Maria Camila Nardini Barioni (UFU)
Mirela Cazzolato (University of São Paulo)
Mirella Moro (Universidade Federal de Minas Gerais)
Renata Galante (Universidade Federal do Rio Grande do Sul)
Renato Fileto (UFSC)
Ricardo Torres (Unicamp)
Ronaldo Mello (Universidade Federal de Santa Catarina)
Sergio Lifschitz (PUC-Rio)
Sergio Manuel Serra da Cruz (Universidade Federal Rural do Rio de Janeiro)
Tiago França (UFRRJ)
Valéria C. Times (UFPE - Universidade Federal de Pernambuco)
Vania Bogorny (Universidade Federal de Santa Catarina)
Vania Vidal (Universidade Federal do Ceará)

Table of Contents (Thesis and Dissertation Workshop)

Um Framework Para a Construção de Chatbots de IQA Baseado em Padrões Sobre Bases de Conhecimento de Domínio Fechado

Caio Viktor da Silva Avila[1], **Vania Maria Ponte Vidal**[1]

[1]Programa de Mestrado e Doutorado em Ciência da Computação (MDCC)
Universidade Federal do Ceará (UFC)
Campus do Pici – Bloco 910 – 60.455 – 760- Fortaleza– CE – Brasil

caioviktor@alu.ufc.br,vvidal@lia.ufc.br

Nível: Mestrado
Ingresso: Março 2018
Previsão de Término: Março 2020
Etapas já Concluídas: Revisão Bibliográfica, Definição do Problema, Conclusão de Exame de Qualificação, Defesa de Poposta
Defesa da Pré-Proposta: Maio 2019
Defesa da Proposta: Junho 2019

Abstract. *In this work we present CONQUEST, a framework that automates much of the construction process of chatbots for the task of template-based Interactive Question Answering on closed-domain knowledge bases. CONQUEST has a flexible question classification mechanism capable of addressing the problem of linguistic variability and uses automatically generated clarification dialog to address the ambiguity and lack of parameters in the question. With CONQUEST, the developer invests his time only in creating the questions supported by the systems, leaving the message control, natural language processing and interpretation of the question to the framework.*

Resumo. *Neste trabalho apresentamos CONQUEST, um framework que automatiza grande parte do processo de construção de chatbots para a tarefa de Interactive Question Answering baseado em padrões sobre bases de conhecimento de domínio fechado. CONQUEST possui um mecanismo flexível de classificação de questões capaz de tratar o problema da variabilidade linguística, além de usar diálogo de clarificação gerado automaticamente para tratar a ambiguidade e a ausência de parâmetros na questão. Com CONQUEST, o desenvolvedor investe seu tempo apenas na criação das questões suportadas pelos sistemas, deixando o controle de mensagens, processamento de linguagem natural e interpretação da questão para o framework.*

Keywords: Interactive Question Answering, ChatBot, Linked Data.

Tabela 1. Lista de publicações.

Título	Conferência
MediBot: An Ontology based Chatbot for Portuguese Speakers Drug's Users	Proceedings of the 21st International Conference on Enterprise Information Systems - Volume 1: ICEIS
MediBot: Um chatbot para consulta de riscos e informações sobre medicamentos	Workshop de Ferramentas e Aplicações (WFA) 2019, XIX Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS)

1. Introdução

Recentemente, o uso de tecnologias do *Linked Data* para a tarefa de integração de fontes de dados em ambientes empresarias vem se popularizando, tornando possível a integração de fontes de diferentes setores, origens, formatos e vocabulários em uma representação única, uniforme e semanticamente integrada [Frischmuth et al. 2012]. Com o uso do *Linked Data*, os dados são disponibilizados como bases de conhecimento (KBs, *Knowledge Bases*) representadas por um vocabulário comum definido por uma ontologia *OWL*, o que permite que as múltiplas fontes heterogêneas sejam acessadas simultaneamente em uma maneira sem costuras através de consultas na linguagem *SPARQL* [Heath and Bizer 2011].

Contudo, a criação de consultas *SPARQL* é uma tarefa desafiadora. Deste modo, exigindo do usuário conhecimentos técnicos sobre as regras de construção da linguagem e sobre o esquema da ontologia sendo consultada.

Uma possível solução para tal problema é o desenvolvimento de sistemas de *Question Answering* (QA). Segundo [Diefenbach et al. 2018], sistemas de QA sobre bases de conhecimento (QA_KB, *Question Answering over Knowledge Base*) tem como objetivo achar no KB a informação solicitada via linguagem natural (NL, *Natural Language*) pelo usuário. Ainda segundo [Diefenbach et al. 2018], a resolução de questões complexas, questões com mais do que um *graph pattern* básico, são um dos principais desafios para este tipo de sistema.

Uma alternativa para o tratamento de questões complexas são os sistemas de QA baseados em padrões (*template-based QA*). Sistemas baseados em padrões tentam classificar a questão do usuário em NL para padrões em linguagens formais de consultas, possuindo *slots* a serem preenchidos por parâmetros informados pelo usuário, onde um padrão pode capturar a estrutura arbitrariamente complexa de uma questão [Diefenbach et al. 2018]. No entanto, estes sistemas possuem algumas limitações, tais como os problemas da (1) classificação inconclusiva (ambiguidade); (2) variabilidade linguística e (3) parâmetros ausentes na questão. Em 1, o sistema não é capaz de classificar a questão para um único padrão, onde o nível de confiança pode ser baixo para todos os padrões ou igualmente alto para mais de um. Em 2, o sistema é capaz de classificar uma questão, apenas quando esta é apresentada de maneira semelhante ao padrão. Por fim, em 3, o padrão para o qual a questão foi classificada exige a entrada de parâmetros (valores para os *slots*) ausentes na questão inicial.

Neste trabalho, para tratar os problemas de ambiguidade e parâmetros ausentes, é recorrido ao uso do *feedback* do usuário, gerando assim sistemas de *Interactive Question Answering* (IQA). IQA é um campo de pesquisa que tem emergido na interseção entre sistemas QA e *Dialog Systems* (DS), e que permite ao usuário achar as respostas para suas perguntas em uma maneira interativa, permitindo ao sistema realizar perguntas ao usuário, sendo assim capaz de solucionar ambiguidades e adquirir informações necessárias para o processamento da consulta [Konstantinova and Orasan 2013].

Como interface para o sistema, neste trabalho é utilizada outra tecnologia de *Natural Language Interfaces* (NLI) que vem se popularizando, os *chatbots*. *Chatbots* são agentes computacionais que servem como interface de usuário em NL para provedores de dados e serviços [Dale 2016]. Os *chatbots* podem ser disponibilizados por meio de

aplicativos de *instant messengers* (e.g., *Telegram*, *Facebook Messenger*, *Skype*, etc.), dispensando a necessidade da criação de uma interface de usuário e treinamento para seu uso, aproveitando a infraestrutura e a base de usuários dos serviços já consolidados.

Assim, neste trabalho, é proposto CONQUEST (*Chatbot ONtology QUESTion*), um *framework* para a criação de *chatbots* para a tarefa de IQA baseada em padrões sobre bases de conhecimento ontológicas de domínio fechado. Como principais contribuições deste trabalho, temos: (1) A definição de uma arquitetura de *chatbots* para tarefa de *template-based IQA*; (2) a construção de um classificador em *machine learning* (ML) que aprenda novas formas para a realização de uma mesma questão, tratando o problema da variabilidade linguística; (3) um algoritmo para a geração automática de diálogos para a desambiguação da classificação de questões e para a solicitação de parâmetros ausentes; e por fim, culminando no (4) desenvolvimento de uma ferramenta que facilite o processo de criação de *chatbots* para a tarefa de IQA sobre domínio fechado.

2. *Framework CONQUEST*

Neste trabalho, é apresentada a arquitetura do *framework CONQUEST* (Figura 1), uma arquitetura que engloba todos os passos necessários para a atividade de IQA baseada em padrões via *chatbots*.

Em *CONQUEST*, o principal artefato concernente à tarefa de QA a ser produzido manualmente pelo desenvolvedor é o conjunto de *Question Answering Itens* (QAI) suportados pelo sistema. Um QAI representa um padrão (*template*) de questão cujo sistema é capaz de responder, onde cada QAI possui *slots* que serão preenchidos com informações da questão do usuário, as chamadas variáveis de contexto (CV, *Context Variables*). Uma QAI é definida como:

$$QAI_{01} = ([QP_1, QP_2, ..., QP_n], SP, RP)$$

- QP_k: Padrão de Questão (*Question Pattern*) em NL associado a questão. Onde $1 \leq k \leq n$.
- SP: Padrão de Consulta SPARQL (SPARQL query Pattern) utilizado para consultar as informações na KB.
- RP: Padrão de Resposta (Response Pattern) em NL apresentado ao usuário.

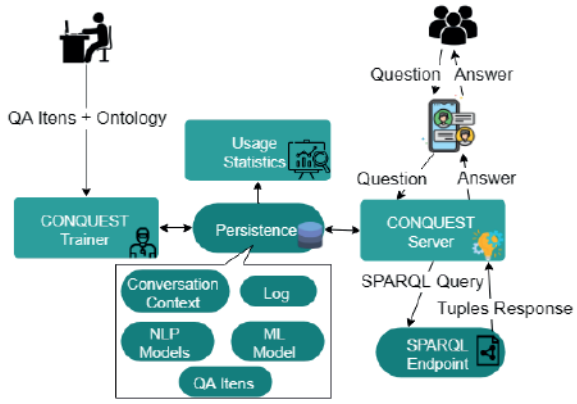


Figura 1. Arquitetura do *framework CONQUEST*.

O componente *CONQUEST Trainer* (Figura 2) tem como entrada o conjunto de QAIs e a ontologia . Este componente é responsável por "treinar" o *chatbot* para executar a tarefa de IQA, armazenando os modelos, índices "aprendidos" no componente *Persistence* que atuará como a "memória" do *chatbot*. Esta etapa é executada *offline* com o seguinte *workflow*:

1. O *Ontology Manager* constrói os índices de classes e propriedades existentes na ontologia;
2. Para cada QAI o *QA Item Manager*:
   (a) Recupera as CVs;
   (b) Confere se as CVs utilizadas nas QPs, SP e RP são compatíveis;
   (c) Realiza o *parsing* da consulta SPARQL SP, avaliando os tipos (*string*, numérica ou data) e a propriedade e classe associada a cada CV;
   (d) Constrói a representação vetorial (QV, *Question Vector*) de cada QP. Onde $QV = VS$ (*Vector Sentence*, vetor representando a sentença QP obtido pelo módulo de NLP) $+CVec$ (*Context Vector*, representação vetorial das classes de instâncias que podem preencher as CV de QP);
   (e) Para cada QP, constrói exemplos de sentenças que serão usadas para treinar o modelo de reconhecimento de entidade nomeadas (NER, *Named Entity Recognition*). Isto é feito ao substituir os marcadores das CVs por valores que estas podem assumir de acordo com o KB.
3. *NER Trainer* treina o modelo NER com base no KB;
4. O *Machine Learning Classifier* treina o classificador de questões com base nos QVs;

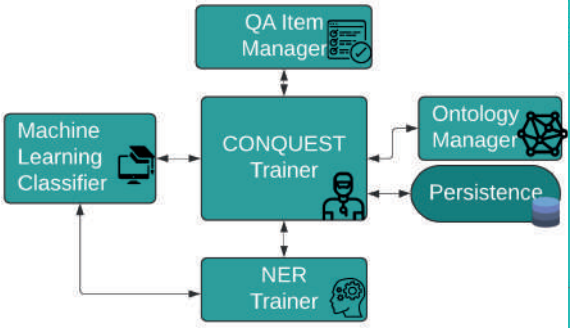


Figura 2. Arquitetura CONQUEST Trainer.

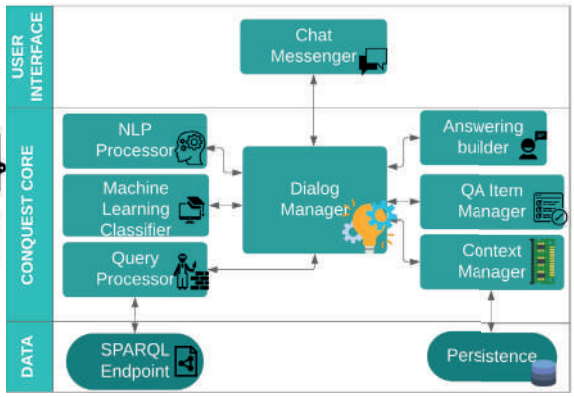


Figura 3. Arquitetura CONQUEST Server.

O componente *CONQUEST Server* (Figura 3) tem como entrada a "memória' armazenada na *Persistence*, o acesso ao *SPARQL Endpoint* do KB e as credenciais de acesso à *API* do *Instant Messenger*. Este componente é responsável por processar a entrada e a saída do *chatbot*, interpretar a questão do usuário, gerenciar o fluxo da interação e consultar o KB, atuando como o "cérebro" do *chatbot* sendo instanciado. A arquitetura apresentada na figura 3 representa a arquitetura de *chatbots* para IQA baseados em padrões proposta neste trabalho, tendo o seguinte *workflow* executado *online*:

1. O módulo *Chat Messenger* recebe a questão Q em NL via *API* do *Instant Messenger*;

2. O *NLP Processor* realiza o parse sintático de Q e o reconhecimento de possíveis valores para CVs com o uso do NER (criando $CVec$). Por fim, realiza a criação da representação vetorial da questão (VP);
3. O *Machine Learning Classifier* classifica a questão para as QAIs (probabilidade), onde $QV = VP + CVec$;
4. Caso a confiança da melhor opção seja baixa ou próxima das demais, apresenta lista de candidatos mais prováveis ao usuário (preenchendo QP com os valores das CVs) solicitando-o a interpretação correta;
5. Caso falte alguma CV, *Answering Builder* solicita ao usuário;
6. O *Query Processor* constrói e executa a consulta SPARQL, preenchendo SP com valores das CVs;
7. *Answering Builder* constrói e apresenta o resultado, preenchendo RP com as variáveis de retorno;
8. Salva o *log* das interações e o *Chat Messenger* envia a mensagem ao usuário.

CONQUEST utiliza um mecanismo baseado em ML para a classificação de questões. O sistema armazena o histórico de uso, utilizando as questões realizadas como um novo conjunto de treinamento para o modelo, permitindo assim que o sistema adapte-se ao uso, passando a dispensar o uso do diálogo de desambiguação. Além disso, tal abordagem permite o aprendizado de novas variações para as questões. Por fim, o módulo final, *Usage Statistics* permite que o desenvolvedor acompanhe o uso do *chatbot*, tendo acesso a informações sobre o uso do *chatbot*.

3. Trabalhos Relacionados

A abordagem apresentada em [Owda et al. 2007] combina técnicas de agentes conversacionais orientados a objetivos e árvores de conhecimento, tendo como objetivo prover a capacidade de desambiguação e conversação sobre o domínio para ambas, questões factuais e complexas. Como limitações, o sistema necessita que o desenvolvedor construa manualmente os *scripts* de interações com o usuário, além de utilizar um mecanismo simples de casamento de padrões, exigindo que a consulta do usuário seja emitida de uma maneira semelhante ao padrão.

Em [Quarteroni and Manandhar 2009] os autores apresentam *YourQA*, um sistema QA de domínio aberto com interface de diálogo baseada em *chatbots*. *YourQA* utiliza o motor de pesquisa do *google* como fonte de conhecimento e utiliza AIML para a definição dos diálogos. Como limitações, o sistema não é capaz de tratar questões complexas. Além disso, o sistema apresenta a resposta como uma lista de trechos que contenham informações relevantes para a pergunta do usuário, pecando no aspecto concisão.

OntBot, apresentado em [Al-Zubaide and Issa 2011], é um *chatbot* baseado em ontologia. *OntBot* responde perguntas em NL sobre a ontologia de domínio em uma maneira conversacional, dando suporte a perguntas complementares. O sistema armazena a ontologia como um banco de dados relacional, gerando automaticamente o esquema da base relacional e os mapeamentos da ontologia para o novo meio de armazenamento. Como principal limitação, o sistema apresenta a necessidade de representação intermediária da fonte de conhecimento.

Ao melhor de nosso conhecimento, *CONQUEST* é o primeiro *framework* de IQA para questões complexas que automatiza o gerenciamento do diálogo, com os demais exigindo que o usuário defina manualmente o fluxo a ser seguido durante uma interação.

CONQUEST alcança isso através do uso de fluxos de interações padrões que são ajustados automaticamente utilizando as informações contidas nas $QAIs$ e na ontologia. Tal abordagem permite que o *chatbot* produzido seja capaz de engajar-se em diálogos para: (1) desambiguar a intenção do usuário e (2) solicitar parâmetros ausentes na consulta, utilizando o SP para identificar os parâmetros e seus tipos. Além disso, ao contrário dos demais sistemas que implementam interfaces próprias, *CONQUEST* produz *chatbots* para os canais já existentes (e.g., *Telegram*) aproveitando sua infraestrutura e disponibilidade.

4. Estudo de Caso

Como um estudo de caso, neste trabalho é apresentado como um padrão de consulta do *chatbot MediBot*[Avila et al. 2019a, Avila et al. 2019b] poderia ser implementado com o uso do *framework* CONQUEST. Durante a fase de treinamento, considerando o padrão de questão: "Qual o preço máximo para um dado medicamento em um certo estado". O desenvolvedor pode dar como entrada a seguinte QAI no formato JSON:

```
'QPs': ["Qual o preço máximo para o medicamento $medicamento no $estado"],
'SP': "SELECT MAX(?precoAux) as ?preco  WHERE{
  ?s a <Medicamento>;
    rdfs:label ?nome;
    <precoMax> ?precoMax.
  ?precoMax <estado> $estado;
    <valor> ?precoAux.
  FILTER(REGEX(?nome,$medicamento,'i'))}",
'RP':{'header': "Este é o maior preço:",'body': "?preco .",'footer':"Ainda deseja algo?"}
```

Tendo como base a ontologia e o parsing de SP, o *framework* será capaz de inferir que a CV \$medicamento é uma *string* e é o valor da propriedade *rdfs:label* da classe *<Medicamento>*. Do mesmo modo, sendo capaz de inferir \$estado como uma *string* da propriedade *<estado>* do *<PrecoMax>*. Deste modo, é gerado o seguinte conjunto de treinamento para o modelo NER:

```
[("qual o preço máximo para o medicamento buscopan no $estado",(39,47,LABEL@MEDICAMENTO)),
("qual o preço máximo para o medicamento $medicamento no ceará",(55,60,ESTADO@PRECOMAX)),...]
```

$CVec$ para QP será um vetor de m dimensões do tipo $(c_1, ..., c_m)$, onde m é o número de pares *propriedade@classe* únicos usados em todas as CVs, onde c_i e c_j que representam *LABEL@MEDICAMENTO* e *ESTADO@PRECOMAX* serão 1, enquanto as demais serão 0. Já o VP de QP será um vetor de n dimensões que representa QP retornado pelo modelo NLP. Deste modo, QV será um vetor de $n+m$ dimensões. Durante a fase de execução, o seguinte diálogo pode ocorrer:

```
U: Diga o preço máximo do buscopan.
B: Você quis dizer: Qual o preço máximo para o medicamento buscopan no estado?
U: Sim.
B: Certo. Mas para qual estado do preço máximo você gostaria?
U: Ceará.
B: Este é o maior preço:  R$ 14,40 . Ainda deseja algo?
U: Não, obrigado.
```

Neste exemplo, o usuário realiza a pergunta de uma maneira consideravelmente diferente ao padrão conhecido. Deste modo, o *chatbot* tenta solucionar a intenção do usuário apresentando o padrão da QP substituindo os valores das CVs encontradas pelo NER na pergunta original. Ao ter a sugestão confirmada, está é adicionada como um novo exemplo para esta QAI (após os valores das CVs serem substituídos por seus identificadores). Contudo, o *chatbot* percebe que ainda falta o valor para a CV \$estado, utilizando assim o tipo inferido da CV para realizar sua solicitação.

Por fim, após substituir os valores das CVs em SP e executá-la no *endpoit SPARQL*, o *chatbot* retorna a resposta seguindo RP. Caso a mesma pergunta seja feita novamente, o *chatbot* terá uma maior confiança na classificação da intenção do usuário, dispensando o diálogo de clarificação:

```
U: Diga o preço máximo do buscopan.
B: Para qual estado do preço máximo você gostaria?
U: Ceará.
B: Este é o maior preço:  R$ 14,40 . Ainda deseja algo?
U: Não, obrigado.
```

5. Conclusões

Neste trabalho apresentamos *CONQUEST*, um *framework* que automatiza grande parte do processo de construção de *chatbots* para a tarefa de IQA baseados em padrões sobre bases de conhecimento de domínio fechado.Os *chatbots* produzidos utilizam o diálogo com o usuário para solucionar os problemas de classificação inconclusiva e a ausência de parâmetros na questão. Além disso, CONQUEST utiliza o aprendizado de máquina para aprender novas maneiras de como uma mesma questão pode ser realizada. Tal característica permite que o *chatbot* adapte-se ao uso.

Referências

[Al-Zubaide and Issa 2011] Al-Zubaide, H. and Issa, A. A. (2011). Ontbot: Ontology based chatbot. In *International Symposium on Innovations in Information and Communications Technology*, pages 7–12. IEEE.

[Avila et al. 2019a] Avila, C. V., Calixto, A., Rolim., T. V., Franco., W., Venceslau, A., Vidal., V. M. P., Pequeno., V. M., and Moura., F. F. D. (2019a). Medibot: An ontology based chatbot for portuguese speakers drug's users. In *Proceedings of the 21st International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 25–36. INSTICC, SciTePress.

[Avila et al. 2019b] Avila, C. V. S., Rolim, T. V., da Silva, J. W. F., and Vidal, V. M. P. (2019b). Medibot: Um chatbot para consulta de riscos e informações sobre medicamentos. In *Anais Estendidos do XIX Simpósio Brasileiro de Computação Aplicada à Saúde*, pages 1–6. SBC.

[Dale 2016] Dale, R. (2016). The return of the chatbots. *Natural Language Engineering*, 22(5):811–817.

[Diefenbach et al. 2018] Diefenbach, D., Lopez, V., Singh, K., and Maret, P. (2018). Core techniques of question answering systems over knowledge bases: a survey. *Knowledge and Information systems*, 55(3):529–569.

[Frischmuth et al. 2012] Frischmuth, P., Klímek, J., Auer, S., Tramp, S., Unbehauen, J., Holzweissig, K., and Marquardt, C.-M. (2012). Linked data in enterprise information integration. *Semantic Web*, pages 1–17.

[Heath and Bizer 2011] Heath, T. and Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136.

[Konstantinova and Orasan 2013] Konstantinova, N. and Orasan, C. (2013). Interactive question answering. In *Emerging Applications of Natural Language Processing: Concepts and New Research*, pages 149–169. IGI Global.

[Owda et al. 2007] Owda, M., Bandar, Z., and Crockett, K. (2007). Conversation-based natural language interface to relational databases. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*, pages 363–367. IEEE Computer Society.

[Quarteroni and Manandhar 2009] Quarteroni, S. and Manandhar, S. (2009). Designing an interactive open-domain question answering system. *Natural Language Engineering*, 15(1):73–95.

Parallel Blocking for Entity Resolution over Heterogeneous Data

Aluno: Tiago Brasileiro Araújo

E-mail: tiagobrasileiro@copin.ufcg.edu.br

Orientador: Carlos Eduardo Santos Pires

E-mail: cesp@dsc.ufcg.edu.br

Universidade Federal de Campina Grande - UFCG
Programa de Pós-Graduação em Ciência da Computação

Nível: Doutorado

Mês e Ano de Ingresso: Março/2016

Mês e Ano Previstos para Defesa: Fevereiro/2020

Etapas Concluídas: Créditos em Disciplinas, Exame de Qualificação, Definição de Problema, Especificação, Análise dos Dados, Referencial Bibliográfico e Publicação de Artigos.

Etapas Futuras: Finalização da Abordagem para Dados em Streaming, Realização de Experimentos e Escrita da Tese.

Abstract. *The Entity Resolution (ER) task emerges as a fundamental step to integrate multiple knowledge bases or to identify similarities between data. To avoid the quadratic cost of ER, blocking techniques are applied as a preprocessing step. In this context, heterogeneous data and large data sources emerge as the main challenges faced by blocking techniques. This work proposes a distributed model for blocking heterogeneous data in the context of large data sources. In addition, we propose novel blocking techniques which can be applied to the proposed model. Based on the experimental results, it is possible to highlight that the novel blocking techniques outperform the start-of-the-art techniques in terms of effectiveness and efficiency.*

Resumo. *A tarefa de Resolução de Entidades (RE) surge como um passo fundamental para integrar múltiplas bases de conhecimento ou identificar semelhanças entre os dados. Para evitar o custo quadrático da tarefa de RE, técnicas de blocagem são aplicadas como uma etapa de pré-processamento. Neste contexto, dados heterogêneos e grandes fontes de dados emergem como alguns dos principais desafios enfrentados pelas técnicas de blocagem. Este trabalho propõe um modelo de execução distribuída para blocagem de dados heterogêneos no contexto de grandes fontes de dados. Além disso, são propostas novas técnicas de blocagem que podem ser acopladas ao modelo em questão. Com base nos resultados experimentais, é possível destacar que as novas técnicas de bloqueio superam as técnicas do estado da arte em termos de eficácia e eficiência.*

1. Introduction and Motivation

Currently, there is an increasing number of information systems producing a large amount of data continuously, such as Web systems (*e.g.*, digital libraries and e-commerce), Social Media (*e.g.*, Twitter and Facebook) and Internet of Things (*e.g.*, mobiles, sensors and devices). These applications have become a valuable source of heterogeneous data [Christen 2012]. Such kind of data presents a schema-free behaviour and can be represented in different formats (*e.g.*, XML, RDF and JSON). Commonly, the data is provided by different data sources and may have overlapping knowledge. For instance, different social media will report the same event and generate mass similar data. Therefore, Entity Resolution (ER) emerges as a fundamental step to support the integration of multiple knowledge bases or identify similarities between entities. The ER task aims to identify records (*i.e.*, entity profiles) from several data sources (*i.e.*, entity collections) that refer to the same real-world entity (*i.e.*, similar entities) [Christophides et al. 2015].

In the context of Big Data, the ER task faces mainly two Vs: *volume*, as it handles a growing number of entities; and *variety*, since different formats and schemes are used to represent the entity profiles [Christophides et al. 2015, Efthymiou et al. 2017]. Blocking techniques and parallel computing can be applied to handle the problems related to the growing volume of data [Araújo et al. 2017]. Blocking techniques aim at grouping similar entities into blocks and perform comparisons within each block, avoiding comparisons guided by the Cartesian product. Parallel processing distributes the computational cost among the various resources (*e.g.*, computers or virtual machines) of a computational infrastructure to reduce the overall execution time of the ER task [Araújo et al. 2017].

Regarding variety, the heterogeneity of the data compromises the blocks generation (by the blocking techniques) since the entity profiles hardly share the same schema. Therefore, traditional blocking techniques (*e.g.*, sorted neighborhood and adaptive window) do not possess satisfactory effectiveness, since the blocking is based on the entity profile schema [Papadakis et al. 2019]. In turn, the variety challenge is addressed by schema-agnostic blocking techniques, which disregard attribute names and consider the values related to the entity attributes to perform blocking [Christophides et al. 2015].

Furthermore, it is possible to highlight three new problems tackled by the ER task: streaming data, incremental processing and noisy data [Christen 2012]. Streaming data is related to dynamic data sources (*e.g.*, from Web Systems, Social Media and sensors), which are continuously updated. When ER receives streaming data, we assume that not all data are available at once. Therefore, ER needs to match the entities as they arrive, also considering the entities already matched previously. On the other hand, incremental ER is related to receive data continuously over time and re-processing only the portion of the matching results (similar entities) that were affected by the data increments. Regarding noisy data, it is commonly characterized by pronunciation/spelling errors and typos in the attribute values of the entities [Liang et al. 2014]. In practical scenarios, people are less careful with the lexical accuracy of the content written in informal virtual environments (*e.g.*, social networks) or when they are submitted to some kind of pressure (*e.g.*, business reports). In the ER context, noisy data directly impact the identification of similar entities, since the spelling difference of their attribute values may determine that two entities, truly similar in the real world, are not regarded as similar by the ER task. Thus, the challenges are strengthened when the ER deals with all these challenges simultaneously.

To address the stated challenges, we propose a parallel model to provide efficiency to the blocking techniques. Furthermore, we also propose novel schema-agnostic blocking techniques capable to incrementally process streaming data and handle noisy data. To the best of our knowledge, there is a lack of blocking techniques for addressing all challenges faced in this work.

2. Methodology

To conduct this scientific research, we divided the whole Ph.D. project in smaller research projects. These research projects address activities related to the following activities: i) theoretical foundation and relevance of the problem to be handled; ii) reading of books, articles and technical reports related to the problem investigated; iii) proposing new approaches and/or methodologies to address RE challenges; iv) definition of hypotheses and planning of an experimental design aiming to evaluate the researched hypotheses; v) formatting and discussion of results obtained in previous activities; vi) listing the interpretations and conclusions resulting from conducting the research; and vii) definition of future work. These activities are applied for all new contribution proposed in our research.

Up to the present moment, in this Ph.D. work, three main research projects were conducted and another one is being developed. The first project is related to define the problem to be addressed during the Ph.D. Based on the bibliography review and the open problems stated for several works, we decide to focus on the ER challenges, as described in Section 1. Moreover, we found the opportunity of improving the efficiency and effectiveness of the state-of-art blocking technique (*i.e.*, Metablocking). As a result of the first project, we published the work [Araújo et al. 2017]. The second project addresses the noisy data challenge. To this end, we propose a novel blocking technique able to handle noisy data without a significant decrease in the effectiveness. This noise-aware blocking technique resulted in the works [Araújo et al. 2018, Araújo et al. 2019]. Regarding the last two research projects, we propose a parallel model to process streaming data incrementally. These works were developed in partnership with the Tampere University (Finland) via a Doctoral Sandwich program. The results achieved during the third research project were described in a paper, which is under review. In the fourth research project, we found the opportunity of improving efficiency and effectiveness of the previously proposed technique, updating the blocking workflow and proposing new algorithms to process the streaming data. The fourth research project should be sent to a journal.

3. Related Work

Several blocking techniques in stand-alone [Papadakis et al. 2019] or parallel [Araújo et al. 2017, Efthymiou et al. 2017, Simonini et al. 2019] modes have been proposed to deal with heterogeneous data. In terms of incremental blocking techniques for relational data sources, [do Nascimento et al. 2018] proposes an approach capable of blocking entities in an incremental way, considering the evolutionary behaviour of data sources to perform the blocking. However, this work does not deal with heterogeneous and streaming data simultaneously.

ER approaches that deal with streaming data, such as [Ma and Yang 2017], do not consider incremental processing and therefore discard the previously processed data. Thus, none of them deal simultaneously with the three challenges (*i.e.*, heterogeneous

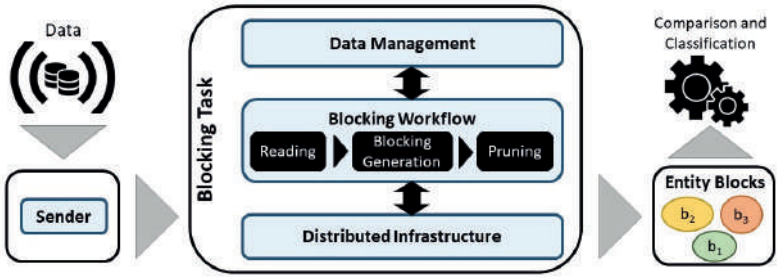


Figura 1. Proposed model.

data, streaming data and incremental processing) addressed by our work. Regarding noisy data, only the work [Simonini et al. 2019] considers the possibility of noise in the data. However, this work does not propose specific strategies to handle noise data. To address this challenge, the noise-aware technique proposed by us guide the whole process of block generation considering the possibility of noise in the data, achieving better results in terms of effectiveness. Overall, in the context of ER, it is possible to detach the lack of works that address challenges related to streaming data and incremental processing simultaneously. In this sense, our work addresses an open research area, which can emerge as a useful schema-agnostic blocking technique for supporting ER in both scenarios.

4. Parallel Model and Blocking Techniques for Heterogeneous Data

To address the stated challenges, we propose a parallel model to minimize the problems related to these challenges. The model is divided into two components: Sender and Blocking Task, as depicted in Figure 1. The Sender component receives the data provided by the data sources in a batch or streaming way. In case of batch data, the Sender reads the data from the data source and sends to the Blocking Task component. For streaming data, the Sender consumes the data and buffers in micro batches and sends to the Blocking Task component. The Blocking component is divided into three layers: data management, blocking workflow and distributed infrastructure. The first layer is responsible to receive and manipulate the data. Thus, the data provided by the Sender is collected and sent to the blocking workflow layer where blocking is performed. The blocking layer is directly connected with the distributed infrastructure layer, which provides all distributed resources (such as the Flink or Spark Streaming engine and nodes/virtual machines) needed to execute the proposed streaming blocking technique.

Notice that the layer blocking workflow can apply different parallel blocking techniques. Therefore, the blocking techniques developed during the Ph.D. research can be used in this model. All the proposed techniques are based on the Metablocking technique [Papadakis et al. 2019], which exploits abstract blocking information to improve the efficiency gains with a minimum impact on the effectiveness. In other words, Metablocking aims to reduce the amount of comparisons generated by each block without discarding comparisons with high chances of resulting in correspondences (*i.e.*, matches). To this end, Metablocking restructures a given set of blocks into a new one that involves significantly fewer comparisons, while maintaining the original level of effectiveness [Papadakis et al. 2019]. This process is called pruning. Initially, a schema-agnostic blocking technique, *e.g.*, token blocking, is applied to block the heterogeneous data. Token blocking extracts tokens (*e.g.*, keywords) from the attribute values of every entity and cre-

ates an individual block for every token that appears in at least two entities. It is important to highlight that blocks generated by token blocking result in a big number of redundant comparisons between entities. For this reason, the blocks generated by token blocking are transformed into a weighted graph, such that each entity is represented by one node and each edge between a pair of nodes infers that the pair of nodes shares at least one block in common. Based on the number of blocks in common between the pair of nodes (pair of entities) linked by the edge, the Metablocking technique defines the weight of each edge (in the graph). Finally, pruning criteria are applied to remove edges with weight below a threshold, which aims to discard comparisons between entities with few chances of being considered a correspondence.

To enhance efficiency of Metablocking, we propose Spark-based Streamlined Metablocking (SS-Metablocking) [Araújo et al. 2017]. The novel approach applies parallel resources (*e.g.*, accumulators and broadcast variables) provided by Spark to assist the execution of the blocking task. Moreover, in this work, we propose the *Global Weighted Node Pruning* (GWNP), a novel pruning algorithm that evaluates globally and locally the graph in order to improve the effectiveness of the approach, without compromising the efficiency. The novel approach is divided into three steps: block filtering, preprocessing, and metablocking. The SS-Metablocking receives as input the blocks generated by a schema-agnostic blocking technique (for instance, token blocking). In the block filtering step, the blocks that have a high cardinality are discarded. The preprocessing step formats the data in order to generate the input for the metablocking step. In the metablocking step, the blocking graph is generated and the GWNP pruning algorithm is applied. As a result, a pruned graph is generated, containing the entities to be compared in the ER task.

To address the problems related to noisy data, we propose the NA-BLOCKER (Noise-aware Schema-agnostic Blocking for Entity Resolution) technique [Araújo et al. 2018, Araújo et al. 2019]: a novel schema-agnostic blocking technique capable of tolerating noisy data to extract information regarding the schema from the data (*i.e.*, group similar attributes based on the data) and enhance the quality of the generated blocks. To this end, the NA-BLOCKER applies Locality Sensitive Hashing (LSH) in order to hash the attribute values of the entities and enable the generation of high-quality blocks (*i.e.*, blocks that contain a significant number of entities with high chances of being considered similar/matches), even with the presence of noise in the attribute values.

To address the streaming data challenges, we propose a novel schema-agnostic blocking technique capable to incrementally process streaming data. Unlike previous parallel Metablocking approaches [Efthymiou et al. 2017], the proposed blocking technique applies a different workflow to reduce the number of MapReduce jobs and improve efficiency. Besides, the proposed workflow does not interfer negatively on the effectiveness results since the block generation is not modified. Furthermore, in this work, we also propose two strategies to improve the efficiency of the blocking technique: attribute selection and top-n neighborhood strategies. Attribute selection aims to discard useless attributes from the entities, enhancing efficiency and minimizing memory consumption. Useless attributes can be understood as attributes that generate tokens that will hardly assist the blocking task to find similar entities. Top-n neighborhood strategy maintains only the "n"most similar entities (*i.e.*, neighbor entities) of each entity, enabling the proposed technique to deal with large data sources.

5. Main Results

In this section, we will present the main results achieved by SS-Metablocking, NA-BLOCKER and the proposed blocking technique for streaming data in terms of efficiency and effectiveness. To measure the effectiveness of blocking, we use: i) Pair Completeness ($PC = \frac{|M(B')|}{|M(D_1,D_2)|}$) that estimates the portion of matches identified, where $|M(B')|$ is the number of duplicate entities in the set of pruned blocks B' and $|M(D_1, D_2)|$ is the amount of duplicate entities between D_1 and D_2; ii) Pair Quality ($PQ = \frac{|M(B')|}{||B'||}$) that estimates the executed comparisons that result in matches, where $||B'||$ is the number of comparisons to be performed in the pruned blocks; iii) Reduction Ratio ($RR = 1 - \frac{||B'||}{||B'_{state-of-art}||}$) that estimates the comparisons avoided in B' (*i.e.*, $||B'||$) with respect to the comparisons produced by the baseline technique (*i.e.*, Metablocking). PC, PQ and RR take values in [0, 1], with higher values indicating a better result. It is important to highlight that schema-agnostic blocking techniques (even the state-of-the-art techniques) yield high recall (*i.e.*, PC), but at the expense of precision (*i.e.*, PQ) [Simonini et al. 2019].

Related to SS-Metablocking [Araújo et al. 2017], we can highlight the better results regarding PQ and RR metrics without decrease the value of PC when compared with Metablocking (the competitor). The most rigorous pruning proposed in GWNP provides a reduction of more than 4,000,000 comparisons when compared with the algorithm applied by Metablocking. Thus, the GWNP achieved better results regarding PQ and RR metrics due to the both metrics are directly influenced by the amount of comparisons resulted by the pruned graph. Regarding execution time, the SS-Metablocking outperformed Metablocking for all variations of the number of nodes (virtual machines).

As described in [Araújo et al. 2019], the NA-BLOCKER technique outperforms the BLAST technique (the competitor) in terms of effectiveness for all data sources with noisy data. The most significant result achieved by NA-BLOCKER was in terms of PQ. In this case, it achieved an average pair quality (considering all pairs of datasets) two times better than the BLAST technique, even in scenarios without noisy data. Regarding efficiency, BLAST achieves better results than NA-BLOCKER. These results are already expected, since the proposed technique added new steps to the blocking task and, consequently, requires more time to block the entities. On the other hand, it is important to highlight that NA-BLOCKER achieved better results compared to BLAST regarding the *aggregate cardinality* (*i.e.*, $||B'||$). In other words, NA-BLOCKER requires less comparisons to be executed in the ER task.

Regarding the proposed blocking for streaming data, it achieved exactly the same results of Metablocking (the competitor) in terms of effectiveness (*i.e.*, PC, PQ and RR). It occurs due to the proposed technique applies the same strategy to generate the blocks. The focus of this work is to propose a technique able to block streaming data efficiently. For this reason, the workflow of the Metablocking was updated and strategies (*i.e.*, attribute selection and top-n neighborhood strategies) to minimize the resource (*e.g.*, memory and CPU) consumption were applied. Thus, based on the preliminary experiments, the proposed technique outperforms Metablocking in terms of efficiency.

6. Conclusion and Schedule

The main objective of this research is to propose a distributed model for agnostic blocking techniques. Thus, up to the present moment, three new agnostic blocking techniques have

been developed, which can be applied to the model. Such blocking techniques seek to achieve better results related both to the efficiency of the blocking execution and to the efficiency of the blocks generated when compared to other state-of-the-art techniques. Moreover, the proposed techniques enable processing of streaming and noisy data.

Regarding the schedule of our Ph.D. research, the most part of the activities were concluded, such as research problem definition, credit compliance (disciplines), definition and development of the parallel model, development of novel blocking techniques, writing and submission of scientific articles and evaluation of the proposed artefacts. Since the thesis will be presented in February of 2020, the last activities are related to concluding the blocking technique for streaming data (which is under development), the application of this technique over a real-world scenario and the writing of the thesis document.

Referências

Araújo, T., Pires, C. E. S., and Stefanidis, K. (2018). Noisy-aware blocking over heterogeneous data. In *Proceedings of the International Semantic Web Conference (PD/Industry/BlueSky)*.

Araújo, T. B., Pires, C. E. S., and da Nóbrega, T. P. (2017). Spark-based streamlined metablocking. In *2017 IEEE Symposium on Computers and Communications (ISCC)*.

Araújo, T. B., Pires, C. E. S., Mestre, D. G., Nóbrega, T. P. d., Nascimento, D. C. d., and Stefanidis, K. (2019). A noise tolerant and schema-agnostic blocking technique for entity resolution. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 422–430. ACM.

Christen, P. (2012). *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media.

Christophides, V., Efthymiou, V., and Stefanidis, K. (2015). Entity resolution in the web of data. *Synthesis Lectures on the Semantic Web*, 5(3):1–122.

do Nascimento, D. C., Pires, C. E. S., and Mestre, D. G. (2018). Heuristic-based approaches for speeding up incremental record linkage. *Journal of Systems and Software*, 137:335–354.

Efthymiou, V., Papadakis, G., Papastefanatos, G., Stefanidis, K., and Palpanas, T. (2017). Parallel meta-blocking for scaling entity resolution over big heterogeneous data. *Information Systems*, 65:137–157.

Liang, H., Wang, Y., Christen, P., and Gayler, R. (2014). Noise-tolerant approximate blocking for dynamic real-time entity resolution. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 449–460. Springer.

Ma, K. and Yang, B. (2017). Stream-based live entity resolution approach with adaptive duplicate count strategy. *International Journal of Web and Grid Services*, 13(3):351–373.

Papadakis, G., Skoutas, D., Thanos, E., and Palpanas, T. (2019). A survey of blocking and filtering techniques for entity resolution. *arXiv preprint arXiv:1905.06167*.

Simonini, G., Gagliardelli, L., Bergamaschi, S., and Jagadish, H. (2019). Scaling entity resolution: A loosely schema-aware approach. *Information Systems*.

Um Modelo Computacional Baseado em Aprendizagem Profunda para a Predição da Umidade do Solo

José Soares da S. Neto[1,2], Ticiana L. Coelho da Silva (orientadora)[1], Regis P. Magalhães (co-orientador)[1]

[1]Programa de Mestrado em Computação (PCOMP)
Universidade Federal do Ceará (UFC)
Campus Quixadá - Bloco IV - 63.902-580 - Quixadá - CE - Brasil

[2]Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI)

jneto.soares@alu.ufc.br, {ticianalc, regismagalhaes}@ufc.br

Resumo. *Segundo a Organização das Nações Unidas para Agricultura e Alimentação (FAO), a agricultura é a atividade que mais consome água no mundo. Cerca de 70% de toda a água consumida no planeta é usada para irrigação de lavouras. Na agricultura de precisão o manejo da irrigação consiste na determinação da quantidade de água, quando e com que frequência a irrigação deve acontecer. A irrigação de precisão considera a umidade do solo que é estimada utilizando o potencial mátrico coletado por vários tensiômetros instalados em diferentes profundidades, coletar esses dados não é economicamente viável para pequenos e médios produtores agrícolas. Portanto, com o objetivo de auxiliar na elaboração de sistemas de irrigação inteligentes mais baratos, este trabalho propõe um modelo baseado em aprendizagem profunda para a predição da umidade do solo através de dados meteorológicos, tornando-se uma solução economicamente mais viável.*

Palavras-chave: *Aprendizagem Profunda, Umidade do Solo, Agricultura de Precisão.*

1. Introdução

Em regiões agrícolas semiáridas, onde os recursos hídricos são menores, um regime de irrigação inteligente é fundamental para a produção agrícola. Esse modelo de irrigação não apenas evita o desperdício do recurso hídrico como também impacta no rendimento da cultura e na diminuição de outros problemas ocasionados pelo excesso ou carência de água no solo [Gu et al. 2017].

A irrigação inteligente utiliza um conjunto de dados meteorológicos e do solo para determinar seu agendamento. Dados meteorológicos são relativamente de fácil obtenção, tendo em vista que miniestações de baixo custo podem ser utilizadas para este fim. Neste sentido, o presente trabalho propõe um modelo para predição da umidade do solo utilizando dados meteorológicos, tornando a atividade agrícola mais sustentável do ponto de vista ambiental e a irrigação inteligente economicamente viável.

É possível estimar a umidade do solo com técnicas de aprendizado de máquina, conforme já investigado em [Ahmad et al. 2010], [Coopersmith et al. 2016] e [Braga et al. 2018]. Há um crescimento da utilização de aprendizado de máquina em sistemas de suporte à decisão para a predição de fatores ambientais e de solo

[Chlingaryan et al. 2018]. Além dessas técnicas, uma nova abordagem que está ganhando força é a aprendizagem profunda, uma evolução das redes neurais artificiais e geralmente usada para lidar com modelos não lineares.

Em geral, modelos de aprendizagem profunda apresentam maior eficácia no aprendizado. As redes *Recurrent Neural Network* (RNN) ou, mais especificamente, *Long Short Term Memory* (LSTM) podem ser viáveis para estimar a umidade do solo, já que tal valor depende das condições meteorológicas e estado do solo nos dias ou horas anteriores. Diversas abordagens propõem modelos derivados da LSTM para diferentes tarefas de predição de dados agrícolas [Laptev et al. 2017], [Kamilaris and Prenafeta-Boldu 2018], porém a maioria desses estudos utiliza bases de dados de imagens e se dedica à identificação de plantas daninhas, classificação da cobertura de solo e tipo de cultura, além da contagem de frutos [Kamilaris et al. 2017].

Este trabalho utilizará um conjunto de dados real fornecido pela Embrapa Agroindústria Tropical. A base de dados contém amostras de dados meteorológicos e dados de solo coletados entre os anos de 2016 a 2018 em uma plantação de caju. Com esses dados é possível determinar a curva característica da umidade do solo através do modelo de [Van Genuchten, M Th 1980], este modelo foi escolhido por ser amplamente utilizado por agrônomos da Embrapa. Além disso, a pesquisa irá comparar a eficácia do modelo de aprendizagem profunda com modelos de regressão linear e de predição para séries temporais, tais como ARIMA e SARIMA, estendendo o estudo feito em [Braga et al. 2018].

A estrutura do trabalho está organizada da seguinte forma. A Seção 2 descreve a fundamentação teórica e a definição do problema da pesquisa. A Seção 3 caracteriza a contribuição deste trabalho e apresenta alguns resultados preliminares. Os trabalhos relacionados são apresentados na Seção 4. Por fim, a Seção 5 expõe as conclusões e as direções futuras.

2. Fundamentação Teórica

Nesta seção são apresentados os principais conceitos para a compreensão deste trabalho, bem como a definição formal do problema abordado.

2.1. Métodos para o Manejo da Irrigação

Existem três tipos de métodos para o manejo da irrigação que determinam quando e quanto irrigar: turno de rega calculado, balanço hídrico e potencial mátrico. Esses métodos são selecionados de acordo com a forma de distribuição de água no cultivo [Frizzone 2017].

O método abordado neste trabalho é o método do potencial mátrico que determina o momento de irrigar monitorando o potencial mátrico atual $(\psi_{m,a})$ em uma profundidade z. Este dado do solo refere-se a interação entre a matriz do solo e água [Frizzone 2017]. Em [Campos et al. 2019] os autores afirmam que podemos considerar o (ψ_m) como uma função contínua da umidade do solo que existe apenas em solos não saturados. Os pesquisadores demonstram a relação da umidade do solo Θ calculada em função do potencial mátrico (ψ_m) através da equação proposta por [Van Genuchten, M Th 1980].

Por esse método é possível calcular a Irrigação Real Necessária (IRN) monitorando a umidade do solo Θ em várias profundidades z_i pela Equação 1. Cada variável da equação é explicada na Tabela 1.

$$IRN = (\Theta_{fc,z_1} - \Theta_{cr,z_1}) \cdot z_1 + (\Theta_{fc,z_2} - \Theta_{c,z_2}) \cdot z_2 + (\Theta_{fc,z_3} - \Theta_{c,z_3}) \cdot z_3 \quad (1)$$

Variável	Descrição
Θ_{cc}	Umidade do solo na capacidade do campo em base de volume
Θ_{cr}	Umidade crítica do solo ou umidade ideal para iniciar a irrigação
Θ_a	Umidade atual do solo
z_i	i-ésima profundidade do sistema radicular

Tabela 1. Variáveis do solo para cálculo do IRN

2.2. Modelos de Aprendizagem Profunda

Uma série temporal é um conjunto de dados indexado por datas. Diferentemente de outros tipos de dados, as amostras de séries temporais aparecem em uma determinada ordem e não são independentes uma das outras. Os dados utilizados por este trabalho são séries temporais.

Este trabalho emprega modelos de aprendizagem profunda para prever, a partir de dados anteriores da série, o próximo valor. Os dados anteriores, aqui mencionados, remetem aos valores de umidade do solo dos últimos dias. O modelo deve prever o valor da umidade do solo do dia atual. O modelo LSTM foi proposto em [Hochreiter and Schmidhuber 1997] e é uma variante da RNN. Uma unidade LSTM é composta de três portas multiplicativas, que controlam as proporções de informação a serem esquecidas e passam para a próxima etapa de tempo. Para muitas tarefas de predição é benéfico ter acesso a contextos passados (à esquerda) e futuros (à direita). No entanto, o estado oculto da LSTM toma informações apenas do passado, sem saber nada sobre o futuro. Uma outra solução é a LSTM bidirecional (BLSTM), que consiste em usar duas camadas regulares de LSTM, cada uma processando a sequência de entrada em uma direção (cronológica e anti-cronológica) e, então mescla suas representações. Ao tratar uma sequência nos dois sentidos, uma BLSTM pode capturar padrões que podem ser perdidos pela versão de ordem cronológica. A Figura 1 representa a arquitetura empilhada de BLSTM a ser investigada neste trabalho.

2.3. Definição do Problema

A implantação de um sistema de irrigação inteligente exige a utilização de uma série de equipamentos relativamente caros, o que a torna inviável para pequenos e médios agricultores, também é condicionada à disponibilização de vários dados de campo fornecidos por especialistas como pesquisadores e engenheiros agrônomos. Diante disso, objetiva-se reduzir os custos desses sistemas, sendo proposto um modelo baseado em aprendizagem profunda, que utiliza dados meteorológicos para a predição da umidade do solo. Tal modelo poderá ser utilizado em conjunto com miniestações meteorológicas de baixo custo, tornando-se uma alternativa viável para pequenas propriedades agrícolas. O problema alvo do estudo pode ser definido da seguinte forma.

Dado um conjunto de dados D, tal que D consiste em uma série temporal contendo o potencial mátrico D_S coletado a partir de sensores no solo, bem como os meteorológicos D_M. A partir de D_S, obtém-se a umidade do solo utilizando o modelo proposto

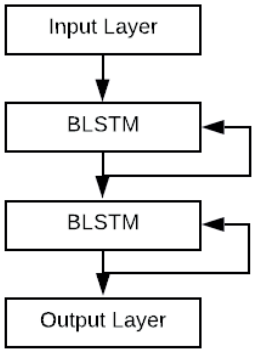


Figura 1. Arquitetura Empilhada BLSTM

em [Van Genuchten, M Th 1980]. No entanto, para obter D_S é necessário um conjunto de tensiômetros instalados em diferentes profundidades no solo, o que pode ser economicamente inviável. Deseja-se construir um modelo preditivo capaz de estimar a umidade do solo a partir dos dados meteorológicos D_M coletados na cultura analisada, tais como temperatura máxima e mínima, umidade relativa do ar máxima e mínima, entre outros, uma vez que D_S pode ser facilmente obtido a partir de miniestações climáticas de baixo custo ou estações mantidas pelo INMET[1].

3. Proposta e Experimentação Preliminar

A finalidade deste trabalho é desenvolver um modelo para predição da umidade do solo utilizando séries temporais de dados climáticos e do solo. Bem como, comparar a performance de modelos de aprendizagem profunda com modelos lineares e tradicionais de predição de séries temporais. As etapas empregadas para atingir o objetivo proposto são descritas a seguir. A primeira etapa consiste em realizar o pré-processamento dos dados utilizados para criação do modelo. Algumas técnicas podem ser aplicadas, tais como detecção e remoção de anomalias, tratamento de features cíclicas, dentre outras. Esta etapa encontra-se em fase de conclusão.

Uma vez realizado o pré-processamento dos dados, eles serão utilizados para a criação do modelo preditivo da umidade do solo utilizando a arquitetura LSTM multivariado apresentada na Figura 1. O uso de um modelo baseado em LSTM decorre do benefício de conter uma conexão recorrente, que é uma espécie de "memória" de entradas que permanecem no estado interno da rede, permitindo fazer uso do contexto passado. Isso a torna mais recomendada para predição em séries temporais [Raschka and Mirjalili 2017]. A qualidade do modelo obtido será avaliada através de métricas como *Root Mean Squared Error* (RMSE) e *Mean Absolute Error* (MAE), tais métricas serão utilizadas para comparar o modelo proposto com os modelos baseados em redes neurais mais simples como AdelineGD e com modelos de regressão linear para predição de séries temporais, estes são apresentados em [Braga et al. 2018].

Experimentos preliminares utilizando AutoML (Automatic Machine Learning) do framework de código livre H2O, apresentaram os seguintes resultados: RMSE = 1.524 e

[1] www.inmet.gov.br/

MAE = 1.124 com um ensemble do metamodelo Gradient Boosting Machine (GBM). O resultados são promissores se comparados com o desvio padrão do label que foi computado em 21.034.

Além do desenvolvimento do modelo preditivo, outras contribuições que poderão ser alcançadas é a identificação das features mais importantes para o modelo, a produção de diferentes bases de dados a serem disponibilizadas publicamente no contexto do problema e avaliação do impacto das características das bases de treinamento sobre a eficácia do modelo.

4. Trabalhos Relacionados

Em [Chlingaryan et al. 2018] os autores realizam um levantamento das pesquisas desenvolvidas nos últimos quinze anos sobre a aplicação de técnicas baseadas em Inteligência Artificial na agricultura. Em [Zhu et al. 2018] os autores apresentam um conciso resumo dos principais algoritmos de aprendizagem profunda utilizados em pesquisas na agricultura e ajudam outros pesquisadores a terem uma visão holística dessas técnicas. Em [Liu et al. 2014] é proposto um método para prever a umidade do solo através da integração de dados climáticos com séries temporais utilizando *Extreme Learning Machine* (ELM). O modelo desenvolvido conseguiu prever com precisão a tendência futura da umidade do solo para um pomar de maçãs. Diferentemente, este trabalho propõe a utilização de aprendizagem profunda para determinar a umidade atual em um cultura de caju, com a utilização de dados meteorológicos.

A contribuição para o desenvolvimento de ferramentas de gestão e apoio à decisão para o manejo da irrigação também é objeto de estudo em braga2018timeseries, onde é apresentado um modelo baseado em aprendizado de máquina para a predição do valor da evapotranspiração. Em [Gu et al. 2017] um método de agendamento de irrigação é proposto baseado no estresse hídrico, modelo de irrigação distinto do utilizado neste trabalho, que utiliza o modelo baseado no potencial mátrico. Para isso, os autores se baseiam no modelo RZWQM2 (*Root Zone Water Quality model*) e utiliza um conjunto de dados simulados. Os próprios autores enfatizam a importância de testar o modelo preditivo em um cenário do mundo real, como proposto neste trabalho.

5. Conclusões

A solução proposta consiste no desenvolvimento de um modelo preditivo utilizando aprendizagem profunda em séries temporais que permita, a partir de dados meteorológicos, a predição da umidade do solo atual. Vale ressaltar que não foram identificadas na literatura abordagens que comparem diferentes modelos preditivos para estimar tal propriedade do solo e que forneçam uma análise comparativa entre diferentes modelos de predição e aprendizagem profunda.

Como trabalhos futuros pretende-se investigar Extreme Learning Machine e como plano para publicações o objetivo é a submissão de um artigo ao journal COMPUTERS AND ELECTRONICS IN AGRICULTURE.

Referências

Ahmad, S., Kalra, A., and Stephen, H. (2010). Estimating soil moisture using remote sensing data: A machine learning approach. *Advances in Water Resources*, 33(1):69–80.

Braga, D., Coelho da Silva, T. L., Atslands, R., Gustovo, C., Rgis P., M., Paulo T., G., and José A. F., d. M. (2018). Time series forecasting for purposes of irrigation management process. In *SBBD 33rd*, pages 217–222. SBC.

Campos, N. G. S., Gomes, D. G., and Rocha, A. R. (2019). Smartgreen: Um framework de internet das coisas para a agricultura de precisão. In *Proceeding of the WTDG X Workshop de Teses e Dissertações do GREat/UFC. Fortaleza-CE, Brazil: WTDG*.

Chlingaryan, A., Sukkarieh, S., and Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and electronics in agriculture*, 151:61–69.

Coopersmith, E. J., Cosh, M. H., Bell, J. E., and Boyles, R. (2016). Using machine learning to produce near surface soil moisture estimates from deeper in situ records at us climate reference network (uscrn) locations: Analysis and applications to amsr-e satellite validation. *Advances in water resources*, 98:122–131.

Frizzone, J. . A. (2017). Necessidade de água para irrigação (lecture notes). Disponível em: https://tinyurl.com/yc26lumq. Acessado em : 10/06/2019.

Gu, Z., Qi, Z., Ma, L., Gui, D., Xu, J., Fang, Q., Yuan, S., and Feng, G. (2017). Development of an irrigation scheduling software based on model predicted crop water stress. *Computers and Electronics in Agriculture*, 143:208–221.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kamilaris, A., Kartakoullis, A., and Prenafeta-Boldú, F. X. (2017). A review on the practice of big data analysis in agriculture. *Computers and Electronics in Agriculture*, 143:23–37.

Kamilaris, A. and Prenafeta-Boldu, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70–90.

Laptev, N., Yosinski, J., Li, L. E., and Smyl, S. (2017). Time-series extreme event forecasting with neural networks at uber. In *International Conference on Machine Learning*, number 34, pages 1–5.

Liu, Y., Mei, L., and Ooi, S. K. (2014). Prediction of soil moisture based on extreme learning machine for an apple orchard. In *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*, pages 400–404. IEEE.

Raschka, S. and Mirjalili, V. (2017). *Python machine learning*. Packt Publishing Ltd.

Van Genuchten, M Th (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils 1. *Soil science society of America journal*, 44(5):892–898.

Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., Huang, M., Zhu, Q., Ji, X., Jiang, Y., et al. (2018). Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *IJABE*, 11(4):32–44.

Usando Aprendizagem de Máquina para Predizer a Ocorrência de Aglomerados de Ônibus em Tempo Real

Aluna: Veruska Borges Santos

E-mail: veruska@copin.ufcg.edu.br

Orientador: Carlos Eduardo Santos Pires

E-mail: cesp@dsc.ufcg.edu.br

Universidade Federal de Campina Grande - UFCG
Programa de Pós-Graduação em Ciência da Computação

Nível: Mestrado

Mês e Ano de Ingresso: agosto/2018

Mês e Ano Previstos para Defesa: maio/2020

Etapas Concluídas: Créditos em Disciplinas, Exame de Qualificação, Definição de Problema, Especificação, Análise dos Dados e Referencial Bibliográfico Inicial.

Etapas Futuras: Finalização e Implementação da Especificação, Realização de Experimentos e Escrita da Dissertação.

Resumo. *Atrasos e superlotação de ônibus são algumas das insatisfações diárias dos usuários de ônibus no Brasil. Esses, muitas vezes, são ocasionados pelos aglomerados de ônibus, evento definido por dois ou mais ônibus executando a mesma rota juntos. Devido à natureza estocástica do tráfego, um cronograma estático não é eficaz para evitar este evento e, portanto, ações preventivas e de correção são necessárias. Nesse contexto, os objetivos desta pesquisa são: i) analisar a correlação entre dados de geolocalização de ônibus, situação do trânsito, clima e a ocorrência dos aglomerados; e ii) propor um modelo eficaz de predição de aglomerados de ônibus em tempo real, utilizando a combinação desses dados, redes neurais artificiais e aprendizagem incremental.*

1. Introdução e Motivação

A Associação Nacional das Empresas de Transportes Urbanos (NTU) relatou que a quantidade de passageiros transportados por ônibus no Brasil caiu cerca de 40% nos últimos 30 anos [NTU 2017]. A ineficiência, má qualidade e a falta de confiabilidade neste serviço estão causando a crescente insatisfação dos usuários, que, progressivamente, optam pelo transporte privado. Essa reopção de meio de locomoção, além de gerar prejuízo financeiro para as empresas de transporte público, prejudica a mobilidade urbana, contribuindo para o aumento dos congestionamentos, da poluição e dos acidentes[dos Santos 2018].

Um dos problemas relacionados ao transporte público é o aglomerado de ônibus (em inglês, *bus bunching*), evento constituído por dois ou mais ônibus executando a mesma rota fisicamente próximos, ou seja, separados por um *headway h* (i.e., o intervalo de tempo de chegada em uma mesma parada) muito pequeno [Nair et al. 2014, Moreira-Matias et al. 2016]. Este evento compromete a qualidade do serviço de ônibus e é uma das causas de insatisfações dos passageiros, como o aumento no tempo de espera e de viagem [Verbich et al. 2016], além do desconforto na viagem devido à lotação.

Assim, melhorar a qualidade dos serviços de ônibus e fornecer informações acuradas e em tempo real para os usuários são necessidades urgentes para atrair os passageiros e melhorar a mobilidade urbana. Uma predição eficaz da ocorrência de aglomerados de ônibus e a determinação de seus fatores influentes podem facilitar a tomada de decisão dos planejadores das empresas, a fim de evitar a ocorrência deste tipo de evento.

Dessa forma, considerando dados de geolocalização, cronograma dos ônibus, da situação do tráfego e do clima, os objetivos desta pesquisa são *i)* realizar a integração e a análise desses dados para encontrar possíveis correlações entre eles e a ocorrência de aglomerados de ônibus e *ii)* propor um modelo para predizer, em tempo hábil e de forma eficaz, a ocorrência desses aglomerados.

2. Trabalhos Relacionados

Desde meados do século passado, o problema de aglomerados de ônibus tem sido estudado por pesquisadores e especialistas da área de transporte, quando os autores de [Newell and Potts 1964], por meio de um modelo simplificado, provaram a instabilidade de uma rota de ônibus, ou seja, a tendência dos ônibus aglomerarem. Desde então, surgiram diversos trabalhos relacionados a este problema, nos quais alguns deles focam em determinar as características espaço-temporais e as causas dos aglomerados de ônibus, enquanto outros focam na predição dos aglomerados de ônibus.

Rashidi et. al. [2017] analisaram dados de GPS (*Global Positioning System*) e concluíram que o desvio do cronograma (horários programados) é o fator mais influente para a ocorrência dos aglomerados. Enquanto isso, Arriagada et. al. [2019], usando dados de GPS e APC (*Automatic Passenger Counting*), indicaram que rotas com alta frequência programada (i.e., com muitos ônibus servindo-as), alta demanda de passageiros, alta variabilidade de demanda e paradas localizadas no final da rota são alguns fatores que contribuem para a aumentar a ocorrência desses aglomerados.

Por sua vez, outros trabalhos focaram na predição dos aglomerados de ônibus. Em [Nair et al. 2014], é apresentado um modelo de predição dos aglomerados a partir da predição dos tempos de chegada nas paradas, usando dados históricos e em tempo

real de GPS e o GTFS (*General Transit Feed Specification*). Uma desvantagem desse modelo é o fato de seu resultado de acurácia ser dependente da alta frequência do GPS, ou seja, quando o intervalo de envio de um dado de geolocalização para o outro é de até 20 segundos. Na prática, nem todas as empresas de transporte possuem sistemas de GPS modernos e com alta frequência. Em Curitiba, por exemplo, os veículos das frotas de ônibus são equipados com GPS cuja frequência média de envio é de 30 segundos, embora requisições em tempo real dos dados só possam ser feitas a cada 2 minutos[1].

Em [Moreira-Matias et al. 2016], os autores propuseram um modelo de predição de *headway* em tempo real e, consequentemente, de aglomerados de ônibus com sugestão de ações corretivas a serem tomadas. Utilizando dados de GPS e GTFS aplicados a modelos de regressão linear e rede neural artificial (RNA), os resultados alcançaram uma precisão de apenas 54%. Isto significa que muitos dos eventos classificados como aglomerados foram, na verdade, resultados falso positivos.

No trabalho de [Andres and Nair 2017], também foi proposto um modelo preditivo de aglomerados de ônibus, o qual prevê os *headways* em tempo real, além de sugerir ações de controle para evitar desvio excessivo do intervalo de viagem programado. Os autores avaliaram quatro métodos de aprendizagem de máquina com dados de GPS e GTFS, os quais apresentaram desempenhos semelhantes: RMSE (*Root Mean Square Error*) entre um e dois minutos. A desvantagem identificada nesse trabalho está relacionada ao uso de algumas premissas que não condizem com a realidade, como a da capacidade ilimitada de passageiros nos ônibus e o fato de um ônibus da mesma rota nunca ultrapassar o outro.

Por fim, Yu et. al. [2017] propuseram um modelo de regressão, denominado RVM (*Relevance Vector Machine*), para realizar a predição de *headway* utilizando dados de cartões de passagens. A utilização desses dados limita as predições apenas em paradas de ônibus onde passageiros embarcaram e, em um cenário real, nem todas as paradas têm embarque de passageiros. Além disso, esses dados são privados e dificilmente disponibilizados pelas empresas, um vez que permitem derivar informações de receita.

Dessa forma, ainda há a necessidade de um modelo eficaz para predição de aglomerados de ônibus em tempo real. Assim, esta pesquisa diferencia-se dos trabalhos já existentes ao propor uma abordagem de predição de aglomerados de ônibus composta por uma combinação de modelos, considerando, além da geolocalização dos ônibus, a integração de dados relacionados ao trânsito, como situação da via e clima. Além disso, não serão utilizados dados de APC, pois são dados privados, esparsos (nem toda parada de ônibus há embarque/desembarque de passageiros) e dificilmente disponibilizados pelas empresas. Outra diferença em relação a estes trabalhos apresentados é a atualização do modelo em tempo real com base no seu erro e de maneira incremental. Portanto, a solução aqui proposta será comparada com o modelo de [Moreira-Matias et al. 2016] por apresentar características semelhantes: uso de RNA e aprendizagem incremental. As métricas de eficácia a serem avaliadas são *precision*, *recall* e *F-measure*, enquanto que a eficiência será avaliada por meio do tempo de execução e do horizonte de predição (i.e., quanto tempo antes da ocorrência do aglomerado o modelo consegue prevê-lo).

[1] URBS: `http://transporteservico.urbs.curitiba.pr.gov.br/index.php` (acesso à documentação e a API através de login, solicitado previamente.)

3. Metodologia

Com o desígnio de conduzir esta pesquisa científica, as técnicas a serem utilizadas são, a princípio, a modelagem e a experimentação. Na modelagem, será investigado como os aglomerados de ônibus relacionam-se com os dados de clima, situação da via, GPS e GTFS, além de estudar como gerar um modelo que captura esse relacionamento. Posteriormente, será investigado, por meio da experimentação, a eficiência e eficácia da abordagem proposta e do modelo a ser comparado em alguns cenários. Assim, nesta seção serão apresentados os dados a serem utilizados e as etapas desta pesquisa.

3.1. Dados

Para realizar esta pesquisa, são considerados dados de quatro fontes distintas:

- **GPS**: dispositivo para coleta de dados de geolocalização de veículos em tempo real, contendo, em geral, informações sobre latitude, longitude, rota, código do veículo e horário de envio do dado, disponibilizados diretamente pela empresa de transporte ou por meio de APIs (*Application Programming Interface*);
- **GTFS**[2]: padrão proposto pela empresa Google para representar horários de transporte público planejados e informações geográficas associadas. Em geral, estes dados são produzidos pela própria empresa de transporte e disponibilizados por meio de solicitação direta ou de APIs;
- **Trânsito**: dados coletados (*crowdsourcing*) de aplicativos colaborativos com informações sobre a situação das vias, a exemplo de engarrafamentos, via interditada e ocorrência de acidentes;
- **Clima**: dados relativos à precipitação atmosférica de bairros de cidades brasileiras, coletados a partir de estações automáticas e disponibilizados pela CEMADEN (Centro Nacional de Monitoramento e Alertas de Desastres Naturais)[3].

Uma amostra dos dados já integrados pode ser vista na Tabela 1, contendo um subconjunto das variáveis exploradas. As três primeiras colunas referem-se a atributos extraídos do GPS e GTFS, a quarta e quinta colunas referem-se à variáveis de clima e, por fim, as duas últimas referem-se à variáveis do trânsito. Cada registro desses dados representa as informações de um veículo em uma parada de ônibus por onde ele passou.

Tabela 1. Amostra dos dados integrados de GPS, GTFS, clima e trânsito.

gpsDateTime	stopPointId	tripProblem	precipitation	precipitationTime	streetSituationDateTime	streetAlert
2018-12-05 04:50:02	3425	BETWEEN	0.0	2018-12-05 04:20:00	2018-12-05 09:24:17	HAZARD_CAR_STOPPED
2018-12-05 05:16:52	3425	NO_PROBLEM	0.0	2018-12-05 05:20:00	2018-12-05 09:24:17	HAZARD_CAR_STOPPED
2018-12-05 05:42:22	3425	NO_PROBLEM	0.0	2018-12-05 05:20:00	2018-12-05 09:24:17	HAZARD_CAR_STOPPED
2018-12-05 05:51:54	3425	BETWEEN	0.0	2018-12-05 05:20:00	2018-12-05 09:24:17	HAZARD_CAR_STOPPED
2018-12-05 05:59:36	3425	BETWEEN	0.0	2018-12-05 05:20:00	2018-12-05 09:24:17	HAZARD_CAR_STOPPED

3.2. Etapas da Pesquisa

A condução desse trabalho, visando propor uma solução para o problema investigado, divide-se essencialmente em cinco etapas. A primeira etapa foi dedicada ao levantamento do estado da arte, no qual foram elencados os trabalhos relacionados à predição de aglomerados de ônibus. A segunda etapa diz respeito à análise e integração dos dados

[2] https://developers.google.com/transit/gtfs/

[3] http://www.cemaden.gov.br/mapainterativo/

a serem utilizados. Em seguida, será possível a proposição da abordagem de predição, correspondendo à terceira etapa desse trabalho. A quarta etapa diz respeito à avaliação da abordagem proposta por meio do planejamento e execução de experimentos. A quinta e última etapa corresponde à divulgação dos resultados obtidos por meio da escrita de artigos e da dissertação de mestrado.

4. Solução Proposta

Nesta seção, é descrita a abordagem proposta (Figura 1) para predição de aglomerados de ônibus.

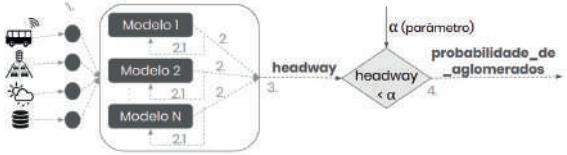


Figura 1. Abordagem proposta para prever aglomerados de ônibus.

Como citado na Seção 3.1, foi planejado utilizar quatro fontes de dados. A partir de dados históricos de GPS, deseja-se recuperar padrões do tráfego, como regiões de maior demanda de passageiros e horários de pico[4] que podem influenciar na ocorrência de aglomerados. Com os dados em tempo real, deseja-se recuperar a localização atual dos ônibus, que, apesar de terem um cronograma a ser seguido, pode variar de acordo com o motorista e com a situação do tráfego, por exemplo. Além disso, deseja-se identificar a situação das vias, coletada a partir de aplicativos colaborativos, que recebem informações em tempo real dos usuários. Os dados históricos também permitirão identificar correlações entre a situação das vias e a ocorrência de aglomerados de ônibus.

Outros dados também utilizados são o cronograma planejado para os ônibus executarem o seu percurso, encontrados no formato GTFS, e os dados históricos e em tempo real do clima da cidade. O cronograma servirá como base para a extração de limiares (como o desvio do *headway*), os quais são empregados para definir a probabilidade de ocorrência de aglomerados. Por sua vez, a situação do tempo, como a ocorrência de chuvas, pode auxiliar na previsão da diminuição da velocidade e consequentes distúrbios no intervalo de viagem entre os ônibus.

Foi realizada a integração de 12 dias de dados de dezembro de 2018 da Cidade X (assim chamada por questões de privacidade), totalizando 1.643.336 registros e 111 variáveis (680MB). Em seguida, foi realizada uma análise de correlação entre essas variáveis usando o teste de Kendall. Tal teste avalia a existência de algum tipo de associação entre um par de variáveis, exibindo o resultado entre -1 e 1, sendo -1 a correlação negativa entre as variáveis, 1 a associação positiva e 0 a ausência de correlação.

O resultado dessa análise pode ser visto na Figura 2. O eixo horizontal representa as variáveis relacionadas ao trânsito, o eixo vertical representa as variáveis de aglomerados de ônibus e os quadrados exibem o valor da correlação entre cada par de variáveis. A

[4]Este padrão foi encontrado ao analisar dados de GPS e GTFS da cidade de Curitiba. Mais detalhes podem ser encontrados em: `https://observablehq.com/@veruska/deteccao-de-aglomerados-de-onibus/2`.

saturação da cor define o nível da correlação: quanto mais próximo de 1 ou -1, maior a intensidade da cor. Foram filtradas da visualização as variáveis com valores constantes (sem variação dos valores não é possível avaliar a correlação) e as variáveis com correlação absoluta abaixo da mediana (0.03). O teste Kendall não revelou correlação significativa entre as variáveis numéricas e a ocorrência de aglomerados de ônibus, evidenciando que um fator por si só não explica a ocorrência desse evento.

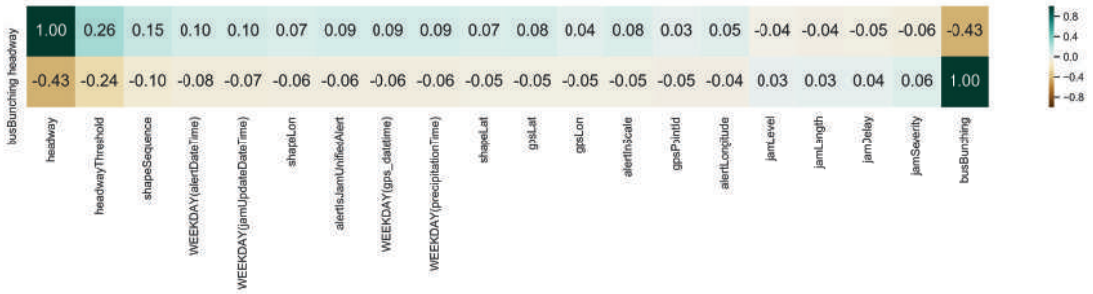


Figura 2. Correlação Kendall entre as variáveis de trânsito e a ocorrência de aglomerados de ônibus.

Os avanços recentes na área de *Machine Learning*, como as Redes Neurais Artificiais, indicam a alta eficácia destes modelos para predizer eventos na presença de grandes quantidades de dados. Com base nessa premissa, espera-se que a combinação de dados heterogêneos aumente a eficácia destes modelos ao predizer a ocorrência de aglomerados de ônibus nas próximas paradas da rota. Diferente do modelo mais simples proposto por [Moreira-Matias et al. 2016], uma *Feedforward Neural Network*, nesta pesquisa será avaliado o uso de CNN (*Convolutional Neural Network*) e LSTM (*Long Short-Term Memory*), porque estas relacionam dados espaciais e temporais, respectivamente. Além disso, a combinação de outros modelos com a RNA, como o *Random Forest*, também será avaliado, baseando-se na premissa dos *ensembles* de que um conjunto de modelos diversos tende a produzir resultados superiores em relação a utilização de um único modelo.

Assim, a abordagem proposta nesta pesquisa tem o objetivo de predizer a ocorrência de aglomerados na próxima parada de ônibus, dada a atual localização do veículo, considerando os atributos oriundos das fontes de dados descritas na Seção 3.1. Este modelo prediz o *headway* na próxima parada entre um ônibus e o seu subsequente da mesma rota. Com base nessa predição, é calculada a probabilidade de ocorrer aglomerado entre esses dois ônibus. Além disso, os modelos são atualizados em tempo real por meio da aprendizagem incremental, utilizando os últimos resíduos (ou seja, a diferença entre o valor predito e o valor real) para recuperar o erro associado.

O fluxograma da abordagem (Figura 1) contém os seguintes passos:

1. Recebe os dados de entrada a cada intervalo de tempo t, converte-os para um formato padrão a ser lido por cada modelo de aprendizagem de máquina;
2. Cada modelo calcula a predição do próximo *headway*;
    2.1. Assim que o valor real é conhecido, atualiza o modelo com base no valor do resíduo do *headway*.
3. Calcula a média ponderada dos *headways* previstos por cada modelo;

4. Com a predição do *headway* e um limiar α, a ser definido com base no GTFS, é calculada a probabilidade final de acontecer aglomerados de ônibus.

Para calcular a probabilidade de ocorrer aglomerados em todas as paradas seguintes à que os ônibus encontram-se, o modelo deve ser executado para cada parada variando apenas as informações da mesma.

5. Considerações Parciais e Próximos Passos

Este trabalho visa propor a criação de um modelo eficaz de predição de aglomerados de ônibus a partir de quatro fontes de dados. Os resultados da predição de aglomerados são de fundamental importância para facilitar a prevenção da ocorrência dos mesmos, contribuindo, assim, com a eficiência e confiabilidade do serviço de transporte público. Atualmente, o projeto encontra-se na fase final de análise dos dados e finalização da especificação do modelo de predição. Não foram encontradas correlações significativas entre as variáveis numéricas e a ocorrência de aglomerados de ônibus que pudessem facilitar o planejamento do transporte público, de modo a evitar os aglomerados. Portanto, um monitoramento automático, que considere a combinação de variáveis, é necessário para lidar com o problema e fornecer informações de predição de forma eficaz e eficiente.

Agradecimentos

Esta pesquisa foi parcialmente financiada pelo INES 2.0, concessão da FACEPE APQ-0399-1.03/17, concessão da CAPES 88887.136410/2017-00 e concessão do CNPq 465614/2014-0.

Referências

Andres, M. and Nair, R. (2017). A predictive-control framework to address bus bunching. *Transportation Research Part B: Methodological*, 104:123–148.

dos Santos, M. B. (2018). Financiamento do custeio do transporte público coletivo urbano. Disponível em: https://www.ntu.org.br/novo/upload/Publicacao/Pub636687203994198126.pdf. Acesso em 30 de dezembro de 2018.

Moreira-Matias, L., Cats, O., Gama, J., Mendes-Moreira, J., and de Sousa, J. F. (2016). An online learning approach to eliminate bus bunching in real-time. *Applied Soft Computing*, 47:460–482.

Nair, R., Bouillet, E., Gkoufas, Y., Verscheure, O., Mourad, M., Yashar, F., Perez, R., Perez, J., Bryant, G., and Transit, M. D. (2014). Data as a resource: real-time predictive analytics for bus bunching. In *Proceedings of the Annual Meeting of the Transportation Research Board, Washington, DC*.

Newell, G. F. and Potts, R. B. (1964). Maintaining a bus schedule. In *Australian Road Research Board (ARRB) Conference, 2nd, 1964, Melbourne*, volume 2.

NTU (2017). Ônibus urbano perde três milhões de passageiros por dia. Disponível em: https://www.ntu.org.br/novo/NoticiaCompleta.aspx?idArea=10&idNoticia=850. Acesso em 29 de dezembro de 2018.

Verbich, D., Diab, E., and El-Geneidy, A. (2016). Have they bunched yet? an exploratory study of the impacts of bus bunching on dwell and running times. *Public Transport*, 8(2):225–242.

Um Processo para Integração de Esquemas em Documentos JSON

Renata J. Padilha[1], **Deise de B. Saccol**[2]

[1 2] Programa de Pós-Graduação em Ciência da Computação

[1]renatajunges52@gmail.com, [2]deise@inf.ufsm.br

Resumo. *Historicamente, a integração de esquemas é algo muito estudado, mas que apresenta, até os dias de hoje, muitas dificuldades decorrentes de inúmeros conflitos e problemáticas. Abordagens voltadas para bancos de dados NoSQL (Not Only SQL) ainda são pouco estudadas, tendo em vista que estes apresentam esquemas implícitos em sua construção. A crescente utilização de documentos JSON (JavaScript Object Notation) mostra a importância de estudos que possam contribuir com a manipulação deste tipo de documento. O presente artigo tem como objetivo especificar um processo para integração de esquemas em documentos JSON. São utilizadas técnicas de similaridade textuais, algoritmo diff e análise da estrutura hierárquica dos documentos. Estas técnicas são aplicadas de forma combinada a fim de determinar se elementos de documentos JSON são equivalentes e se os mesmos podem ser integrados.*

1. Introdução

Com o advento dos bancos de dados NoSQL [Sadalage and Fowler 2012], a manipulação de grandes volumes e variedades de dados se tornou algo eficiente, se comparado aos bancos de dados relacionais. Conforme [Ranking 2019], o banco de dados NoSQL mais utilizado é o MongoDB, que apresenta como principais formatos de documentos XML (*Extensible Markup Language*) e JSON (*JavaScript Object Notation*).

JSON é um formato leve que consiste em uma coleção de pares chave/valor. Por não possuir um esquema definido, apresenta uma estrutura implícita, sendo que diversas pesquisas realizaram estudos voltados para a extração desses esquemas [Machado 2017] [Izquierdo and Cabot 2013]. Outro ponto relevante é a necessidade de se realizar a integração de diferentes esquemas oriundos de documentos JSON.

Tendo como base estudos prévios sobre integração de esquemas voltados para tanto esquemas relacionais quanto XML, este artigo objetiva descrever um processo de integração de esquemas para documentos JSON. São aplicados diferentes métodos, como o algoritmo *diff* e técnicas de similaridade, tais como extração de radicais das palavras, perfil de conteúdo, similaridade sintática e de sinônimos. Além disso, é levada em consideração a estrutura hierárquica dos elementos.

A metodologia adotada é brevemente descrita a seguir: o levantamento bibliográfico se deu através de pesquisa exploratória, usando fontes de pesquisa primárias e secundárias. Este estudo foi sumarizado através de uma exposição qualitativa dos conceitos estudados. O processo de integração foi especificado com base nos conceitos adquiridos neste levantamento e em demandas percebidas na análise de documentos de teste, obtidos na web. O estudo de caso está sendo feito em um conjunto de dados mais amplo, também obtido na web, e sucessivos refinamentos do processo estão sendo realizados. A validação do processo se dará por análise dos resultados obtidos, com foco em medidas de revocação e precisão.

A organização do artigo está dividida da seguinte maneira: primeiramente, na Seção 2 é realizada uma revisão bibliográfica de trabalhos relacionados. A descrição geral do processo de integração de esquemas, assim como o detalhamento de cada fase, pode ser vista na Seção 3. A conclusão é descrita na Seção 4.

2. Trabalhos Relacionados

Considerando os estudos analisados, foi observada a falta de pesquisas voltadas para a integração de esquemas de documentos JSON. Foram encontrados trabalhos voltados para a realização da extração de esquemas de documentos JSON [Machado 2017] e [Izquierdo and Cabot 2013] mas verificou-se que a integração de esquemas para dados semi-estruturados é bastante voltada a documentos XML [Lauri Mukkala and Knuutila 2017]. Logo, os trabalhos de integração de esquemas XML e a extração de esquemas de documentos JSON foram considerados para a análise dos trabalhos relacionados neste artigo.

Em [Izquierdo and Cabot 2013] é proposto a geração de um esquema para os objetos JSON derivados de diferentes serviços de API. Para isto, é realizado um processo que passa por inúmeras fases para descobrir as informações de esquemas de documentos JSON.

A pesquisa de [Lauri Mukkala and Knuutila 2017] descreve uma ferramenta para casamento de esquemas XML, chamada TRC-Matcher. A proposta é de um algoritmo *matcher* híbrido, que utiliza métodos baseados em: dicionário de sinônimos WordNet e abreviações, distância de Jaro-Winkler e perfil de conteúdo. É importante destacar que o TRC-Matcher não usa a estrutura para combinar os elementos.

Um processo para a extração de esquemas em fontes de dados JSON pode ser visto em [Machado 2017]. Por meio da análise dos campos que representam a mesma informação, mas são escritos de maneiras diferentes, o processo faz uma análise de similaridades de caracteres, conhecimento e radicais. As técnicas de similaridade textual utilizadas são: dicionário de sinônimos WordNet com a medida de *Lin*; distância *Levenshtein*; e extrator de radicais Porter. Esta pesquisa de [Machado 2017] levou em consideração apenas as similaridades textuais presentes nos campos dos documentos JSON, enquanto a pesquisa realizada neste artigo, além de considerar as similaridades textuais dos campos, leva em consideração os dados e a estrutura (hierarquia) dos campos.

Os estudos citados anteriormente fazem uso de diferentes métodos para realizar a correspondência de esquemas. Nestes estudos não foram relatados testes com documentos JSON voltados para integração, apenas para a extração. Sendo assim, acredita-se que especificar um processo para integração de esquemas de documentos JSON é uma contribuição relevante, uma vez que a representação integrada dos diversos elementos presentes nos documentos torna sua modelagem mais eficaz e permite o acesso integrado por aplicações.

3. Processo para integração de esquemas de documentos JSON

O processo de integração de esquemas proposto neste trabalho é ilustrado na Figura 1. A separação em quatro fases é essencial para detalhar as atividades e subatividades pertencentes a cada momento. A 1ª fase realiza um pré-processamento dos documentos JSON, extraindo informações para a 2ª e 3ª fase. Em seguida, a 2ª fase analisa as similaridades textuais dos elementos dos esquemas. Por meio da verificação de similaridades presentes nos ancestrais dos elementos candidatos à integração, a 3ª fase é executada. Com base nos elementos equivalentes e os mapeamentos, a 4ª fase gera o esquema integrado.

3.1. Pré-Integração – 1ª Fase

Nesta primeira fase os documentos JSON (com os campos e dados) passam por um pré-processamento. Foi escolhido como entrada o domínio de publicações científicas de bibliotecas digitais como *PubMed*[1] e *Bibsonomy*[2]. É importante salientar que o processo permite comparar esquemas de documentos de *datasets* distintos, mas que pertençam a um mesmo domínio.

[1]https://www.ncbi.nlm.nih.gov/pubmed
[2]https://www.bibsonomy.org/

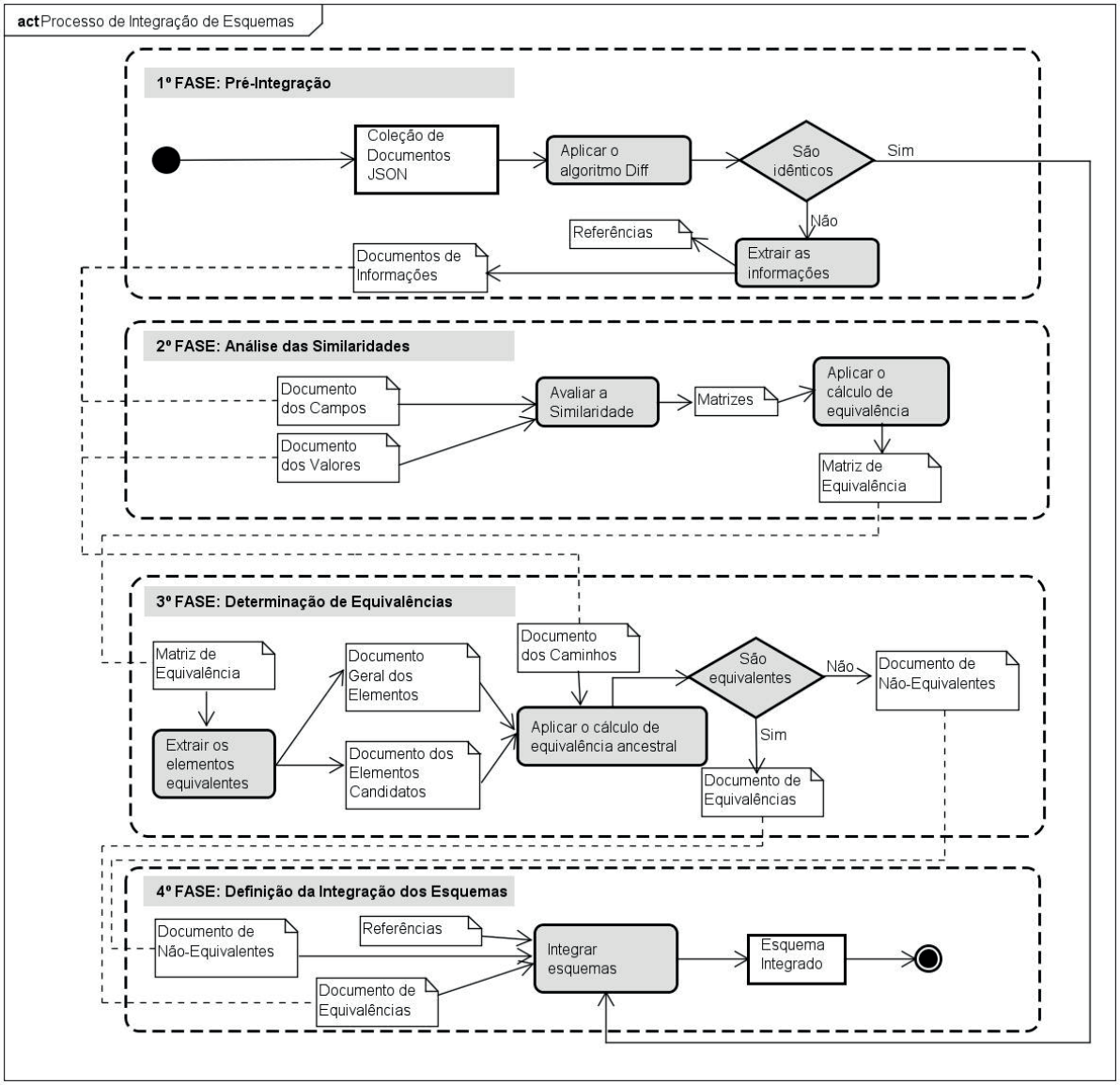


Figura 1. Processo de integração de esquemas de modo geral - Fonte: Autor

A estratégia binária balanceada foi utilizada, onde os esquemas foram comparados em pares, sendo a ordem de escolha conforme a ordem dos documentos dentro da coleção.

A atividade ***Aplicar o algoritmo diff*** compara os nomes e valores dos campos dos arquivos JSON, gerando as diferenças existentes entre eles. Se os documentos forem idênticos irão diretamente para a última fase para serem integrados, mas se não forem idênticos passam para a atividade ***Extrair as informações***, onde é percorrido todo o documento, extraindo informações essenciais. Esta atividade é dividida em: *Extrair os campos*, *Extrair os valores* e *Extrair os caminhos*, onde são coletados todos os campos, valores e expressões de caminhos dos documentos, respectivamente. Para isto, são utilizadas a API Java JSR 353, que processa documentos JSON por meio de um *parser* e a linguagem JSONPath, sendo possível visualizar o caminho que os campos estão inseridos dentro de um documento JSON.

Os artefatos de saída são o *Documento de Informações* e as *Referências*. O primeiro é composto por três arquivos de texto, denominados de *Documento dos Campos*, *Documento dos Valores* e *Documento dos Caminhos*. Todos estes documentos foram gerados a partir de

implementações em Java. As *Referências* contém uma listagem dos campos com seus documentos JSON de origem.

3.2. Avaliação da Similaridade – 2ª Fase

A aplicação de medidas de similaridades textuais é executada nesta fase, tendo como objetivo identificar palavras equivalentes. Quatro técnicas são aplicadas: extração de radicais, análise de caracteres e de conhecimento e aplicação de perfil de conteúdo. Algumas das técnicas foram escolhidas tendo como critério a observação realizada nos estudos de [Didik Dwi Prasetya and Hirashima 2018]. A atividade ***Avaliar a similaridade*** aplica as quatro técnicas para avaliar o quanto duas palavras são equivalentes, resultando em quatro matrizes provenientes de cada técnica. As comparações são realizadas de modo que as palavras sejam testadas "todas com todas". Três subatividades utilizam as palavras provenientes do *Documento dos Campos*, e a subatividade *Analisar o perfil de conteúdo* faz uso do *Documento dos Valores*.

A subatividade *Analisar os radicais* diz respeito à técnica de *Stemming* (extrator de radicais), que é executada com o algoritmo de [Porter 2006]. São realizadas comparações entre os radicais, no caso de total similaridade, o valor gerado é 1 (um), mas se não há nenhuma similaridade, o valor gerado é 0 (zero). A segunda subatividade, *Analisar similaridade de caracteres*, leva em consideração a relação sintática das palavras. A função aplicada foi a de Jaro Winkler [Winkler 1990] que calcula o número de caracteres comuns das palavras. Os valores gerados ficam no intervalo de 0 (zero) a 1 (um).

Em *Analisar similaridade de conhecimento* a semântica das palavras é levada em consideração, onde foi utilizada a ferramenta WordNet [Miller 1995], com a medida de [Wu and Palmer 1994]. Esta utiliza o menor comprimento de caminho entre os conceitos. Para a implementação foi utilizado o *WS4J*[3], considerando os resultados obtidos na medida de Wu Palmer, trazendo valores no intervalo de 0 (zero) a 1 (um). E a última subatividade, *Analisar o perfil de conteúdo* é descrito por [Lauri Mukkala and Knuutila 2017], que se baseia nos padrões que podem ser encontrados nos dados. Depois de adicionado os perfis de conteúdo para cada palavra é realizada uma comparação para determinar se são equivalentes (valor 1) ou não (valor 0). Três análises descritas anteriormente (Porter, Jaro Winkler e WordNet) já foram testadas na linguagem Java, enquanto o perfil de conteúdo foi realizado manualmente.

A atividade ***Aplicar o cálculo de equivalência*** executa cálculos e testes a fim de determinar o grau de similaridade entre os elementos das matrizes resultantes. O método de tomada de decisões AHP (Processo Analítico Hierárquico) é utilizado. Este auxilia no processo para definir os pesos adotados para cada técnica de similaridade textual. Por meio de uma hierarquia de relevância, que pode ser vista em [Saaty 2008], foram atribuídos os pesos 1,2,3 e 4 para as técnicas de perfil de conteúdo, extrator de radicais, similaridade de caracteres e similaridade de sinônimos, respectivamente.

Algumas observações referentes aos valores das matrizes de cada técnica são importantes para a determinação da *Matriz de equivalência*: se qualquer uma das técnicas obtiver valor 1 é considerado equivalente; se todos obtiverem valor 0 é considerado não equivalente; se uma das técnicas obtiver valor no intervalo de 0 a 1, é atribuído o valor 1 (se valor maior que 0,7) ou 0 (se valor menor ou igual a 0,7); em nenhum dos casos, é aplicado a média ponderada, verificando se o valor for maior que 0,5 atribui-se 1, senão 0.

A definição do ponto de corte é por meio de um limiar, que é o fator que separa os elementos relevantes dos irrelevantes. Existem estudos que realizam definições quanto a isto, como pode ser visto em [Juliana B. dos Santos and Wives 2011]. Conforme [Machado 2017],

[3]Wordnet Similarity for Java. https://github.com/Sciss/ws4j

os valores dos pontos de cortes (0,7 e 0,5) foram determinados tendo como base testes realizados com domínios de aplicação semelhantes a este estudo. A saída da 2ª fase é a *Matriz de equivalência*, que foi implementada parcialmente, contendo os resultados das técnicas de similaridade e cálculo de equivalência aplicadas.

3.3. Determinação de Equivalência de Similaridades – 3ª Fase

A determinação de quais palavras são equivalentes é demonstrada nesta fase por meio da avaliação da estrutura hierárquica dos esquemas.

A atividade ***Extrair os elementos equivalentes*** compreende as palavras resultante da *Matriz de equivalência*. O *Documento Geral dos Elementos* lista todos os pares de palavras com seus valores correspondentes (0 ou 1). O *Documento dos Elementos Candidatos* contém os pares de palavras que possuem o valor 1. A atividade seguinte, ***Aplicar o cálculo de equivalência ancestral***, verifica se os ancestrais dos elementos candidatos são equivalentes, determinando se podem ou não serem integrados. Esta atividade é dividida em duas subatividades:

Extrair os elementos ancestrais dos candidatos: Cada uma das palavras do *Documento dos Elementos Candidatos* é analisada e, por meio do *Documento dos Caminhos* dos elementos, o primeiro ancestral (elemento pai) é extraído para outro documento chamado de *Documento dos Ancestrais*. Uma listagem de *Referências* é guardada para determinar, posteriormente, a qual par de palavras pertencem os elementos ancestrais.

Analisar as similaridades: Determina se os pares de palavras contidas no *Documento dos Ancestrais* possuem similaridades. Por meio do *Documento Geral dos Elementos* é verificado se o par de palavras ancestrais possuem valor 0 (não equivalente) ou 1 (equivalente). Utiliza as *Referências* e então, dependendo da análise dos elementos ancestrais, o par de palavras candidatas é armazenada em um *Documento de Equivalências* ou *Documento de Não-Equivalentes*. As aplicações destas atividades foram realizadas manualmente.

3.4. Definição da Integração dos Esquemas – 4ª Fase

A última fase realiza a integração dos esquemas propriamente dita. Os artefatos de entrada são os documentos com as *Referências*, *Documento de Equivalência* e o *Documento de Não-Equivalentes*. O primeiro contém as informações de quais documentos JSON as palavras pertencem, decorrente da 1ª fase. O segundo e terceiro são os pares de palavras que podem e não podem ser integrados. O esquema integrado, composto pelos campos e sua hierarquia, é o artefato de saída desta fase e do processo como um todo. Uma representação gráfica pretende ser implementada para a visualização da integração dos esquemas ser melhorada. Destaca-se que durante todo o processo não há perda de informações dos documentos.

A atividade ***Integrar esquemas*** realiza os mapeamentos e a integração das palavras classificadas como equivalentes. É composta por subatividades: *Construir os Mapeamentos* e *Montar estrutura*. Por meio da lista de *Referências* e dos *Documentos de Equivalências* e *Não-Equivalentes* é construído o *Mapeamento*. Este é um documento de texto que determina, a partir dos pares de palavras equivalentes e não equivalentes, a quais documentos de origem as palavras pertencem. Os mapeamentos são importantes para futuras consultas. Depois o objetivo é consolidar a estrutura, resultando no *Esquema Integrado*, tendo como parâmetros as palavras determinadas como equivalentes, assim como os mapeamentos. A forma como o esquema integrado será representado ainda será definida posteriormente.

4. Conclusões

A integração de esquemas é uma tarefa estudada há muitos anos e que vem sofrendo algumas modificações, devido ao aumento do uso de esquemas para documentos. É relevante destacar que esquemas implícitos, presentes em documentos XML e JSON, são importantes para

se obter uma visão das diferenças estruturais presentes no banco de dados. A abordagem da integração de esquemas mostra a importância de pesquisas voltadas especificamente para tratar os documentos JSON.

O processo de integração de esquemas descrito é composto por algumas fases e diversas atividades e subatividades. Alguns testes já foram realizados manualmente e outros foram implementados na linguagem Java. O período previsto para a defesa da dissertação é em fevereiro de 2020, sendo assim, os testes e o restante da escrita será finalizada até dezembro de 2019. Por meio desta contribuição será possível obter um esquema integrado derivado de inúmeros documentos JSON pertencentes a uma coleção do mesmo domínio de aplicação.

Um artigo foi submetido no periódico IJSEKE (*International Journal of Software Engineering and Knowledge Engineering*), cuja contribuição como coautora neste trabalho faz referência à algumas implementações que realizaram testes de equivalência entre elementos presentes em documentos JSON. O artigo está em fase de alterações após solicitação dos revisores.

Referências

Didik Dwi Prasetya, A. P. W. and Hirashima, T. (2018). The performance of text similarity algorithms. International Journal Of Advances In Intelligent Informatics. Vol. 4, No. 1, Pp. 63-69.

Izquierdo, J. L. C. and Cabot, J. (2013). Discovering implicit schemas in json data. In LNCS, volume 7977 LNCS of ICWE'13, pages 68–83, Berlin, Heidelberg. SpringerVerlag.

Juliana B. dos Santos, Carlos A. Heuser, V. P. and Wives, L. K. (2011). Automatic threshold estimation for data matching applications. Information Sciences, [S.L.], V.181, N.13, P.2685–2699.

Lauri Mukkala, Jukka Arvo, T. L. and Knuutila, T. (2017). Trc-matcher and enhanced trc-matcher - new tools for automatic xml schema matching. University Of Turku Technical Reports, No.13.

Machado, F. T. D. S. (2017). Um processo para extração de esquemas conceituais em fontes de dados json baseado em técnicas de similaridade de texto. Programa De Pós-Graduação Em Ciência Da Computação (Ppgcc). Universidade De Federal De Santa Maria (Ufsm).

Miller, G. A. (1995). Wordnet: A lexical database for english. Communications Of The Acm, [S.L.], V.38, N.11, P.39–41.

Porter, M. (2006). The porter stemming algorithm. http://tartarus.org/martin/PorterStemmer/.

Ranking, D.-E. (2019). Trend popularity. https://db-engines.com/en/ranking_trend.

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. International Jornal Of Services Sciences, [S.L.], V.1, N.1, P.83–98.

Sadalage, P. J. and Fowler, M. (2012). Nosql distilled: a brief guide to the emerging world of polyglot persistence. [S.l.]: Pearson Education.

Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. U.S. Bureau of the CensusStat. Research Div., Rm. 3000-4, Washington, DC20223.

Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. In Proceedings Of The 32nd Annual Meeting Of The Association For Computational Linguistics, Las Cruces, New Mexico.

MOON: An Approach to Data Management on Relational Database and Blockchain

Carlos Sérgio da Silva Marinho [1,2,3], **Leonardo Oliveira Moreira** [2,3], **Javam de Castro Machado** [1,2,3]

[1]Programa de Mestrado e Doutorado em Ciência da Computação
[2]Laboratório de Sistemas e Banco de Dados
[3]Universidade Federal do Ceará, Fortaleza, Brasil

{sergio.marinho,leonardo.moreira,javam.machado}@lsbd.ufc.br

Resumo. *Bancos de Dados Relacionais (BDR) têm sido difundidos por décadas. No entanto, novas tecnologias emergem, como Blockchain. Essa tecnologia possui propriedades relevantes, por exemplo imutabilidade e ausência de terceira parte confiável. Assim, é factível que aplicações que usam BDR migrem parte de seus dados para Blockchains, de modo a se beneficiar dessas propriedades. Essa pesquisa propõe o MOON, uma abordagem hibrida para gerenciar dados em BDR e Blockchain, que recebe consultas SQL.*

Abstract. *Relational Databases (RDB) have been widespread for decades. However, new technologies emerge, such as Blockchain. It has relevant properties, for example, immutability and no trusted third party. Thus, it is feasible for applications that use RDB to migrate part of their data to Blockchains to benefit from these properties. This research proposes the MOON, a hybrid approach to manage data in RDB and Blockchain, which receives SQL queries.*

1. Introduction and Motivation

A Database is a collection of related data. The relational model is widely used in applications from various contexts. This model defines a Database as a collection of one or more relations, which are composed of rows and columns. On the other hand, Blockchain technology provides distributed, reliable, and secure support for large-scale peer-to-peer (P2P) network transactions [Nakamoto 2008]. At Blockchain, there is a decentralized trust entity, without the need for reliable centralized third parties, such as notary offices, and makes it a disruptive technology. Besides, this feature allows network participants not necessarily to trust each other to work rightly [Greve et al. 2018].

The Relational model has been used extensively for decades and was substantial for the popularization of the use of Databases. Currently, many applications continue to be data-oriented [Maenhaut et al. 2015], but alternatives to the Relational model have been solidifying. One of them is Blockchain technology, which can be used for applications to benefit from its properties, such as immutability and irrefutability. The use of Blockchain is advantageous to Relational Databases (RDBs) in aspects such as security and trust-building. However, RDBs usually have higher throughput in writing operations than Blockchains [Chowdhury et al. 2018]. This probably occurs because, unlike Blockchains, RDBs do not implement expensive consensus, such as Proof of Work, which is used by several Blockchains in the literature [Yaga and Mell 2018].

Given this scenario, an application can use datasets that are best suited to the Relational model and others that are most suitable to Blockchain. More inherent data of Blockchain are in contexts in which there is no reliable third party, in addition to having no trust between the nodes of the network or immutability of the data. On the other hand, data that are more adequate to the Relational model are those that are desired to perform more complex queries, with the use of joins or analytics. Thus, the best for those applications may be to use a hybrid approach. Based on the current state-of-the-art, it has been found that there is a shortage of works that propose approaches to hybrid data management, with data storage in RDB and Blockchain. Besides, this work introduces the use of Structured Query Language (SQL) as a way of interacting applications with the approach, which makes access more transparent to the model the data used.

Research questions: (i) What are the advantages of developing an approach that maintains data disjointly in Blockchains and RDBs ?; (ii) given a collection of data, which aspects should be considered for decision-making on how best to store it: Blockchain or RDB?; (iii) how to map relational data to a Blockchain infrastructure?

Hypothesis: (i) An approach for managing data that uses the Relational and Blockchain model can provide the applications: (a) complex queries and data mutability, relational model characteristics and; (b) immutability, transparency and lack of reliable third party, inherent to Blockchain.

Contributions: (i) Develop a data management approach to store data in Blockchain and RDB; (ii) design a relational data mapping for Blockchain; (iii) develop a case study to evaluate the proposed approach.

2. Background

Generically, a Blockchain is a structure in which each block has a set of transactions and indicates the previous block using a hash pointer. A Blockchain network is a P2P network in which each node has a replica of that structure [Nakamoto 2008]. Blockchain technology has some relevant properties. One of them is that it is inherently decentralized and has high availability since it is available even if some of the network nodes are offline. This technology is also auditable because transactions stored in the ledger can be inspected by the network nodes. Besides, the transactions are also irrefutable, since a node can not contest the authenticity of a transaction sent by itself. Finally, transactions published in blockchains are immutable since they can not be tampered [Chowdhury et al. 2018, Greve et al. 2018, Zheng et al. 2017].

3. Proposed Approach

The proposed approach, denominated *an approach to data **M**anagement on relati**O**nal database and bl**O**ckchai**N*** (MOON), aims to manage data in a hybrid way, partitioned between a Blockchain and an RDB. The client applications communicate with MOON through the SQL language, regardless of the data is inserted in a Blockchain or an RDB. There are two advantages of using SQL: (i) not creating a new interaction model for communication with MOON, to use a widely used language known to developers; and (ii) providing transparency, so that it is not necessary to interact differently for each form of persistence used. In this way, there may be stakeholders who interact with the system and do not know how the data is being stored.

The use of SQL introduces another challenge, which is to map to Blockchain the requests made in a language developed for the Relational model. Blockchains are not designed to store relational data, and some change in the structure of this data is required to store it in a Blockchain, preserving its semantic function. The possibility that will be adopted in the proposed solution is exemplified in Figure 1, which is transforming each tuple of the relational model for object notation in JavaScript Object Notation (JSON). Arrow 1 shows a client SQL request to the MOON and arrow 2 indicates the MOON sending a request to the Blockchain network. Arrows 3 and 4 express the returns, respectively, in JSON and table formats.

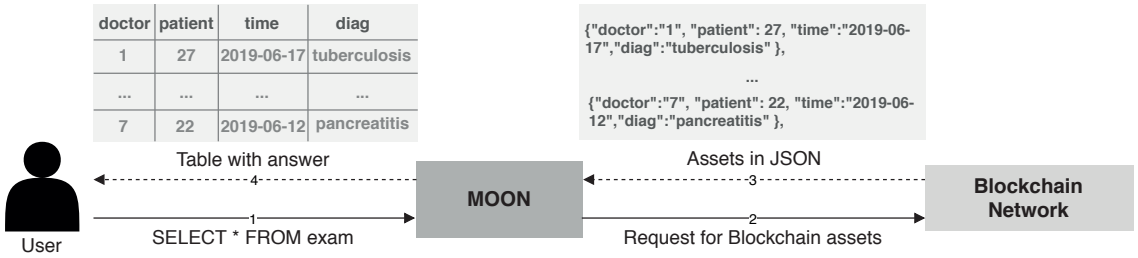


Figure 1. Mapping from Blockchain to the Relational Model

In the architecture presented in Figure 2, five modules were proposed. The User sends SQL requests to the MOON through interaction with the Communication Module. Next, it sends the request to the Scheduler Module, which is responsible for sorting the received transactions and for forwarding them to the SQL Client, if they refer to the RDB. If the request is related to Blockchain, the Scheduler sends it to the Mapping Module, which is responsible for receiving requests from the relational model and sending them to the Blockchain Client, making the necessary changes so that Blockchain Client can communicate with the Blockchain Network. The SQL Client can contain several drivers from different Databases Management Systems (DBMS), such as Oracle and PostgreSQL, and sends received requests to RDBs.

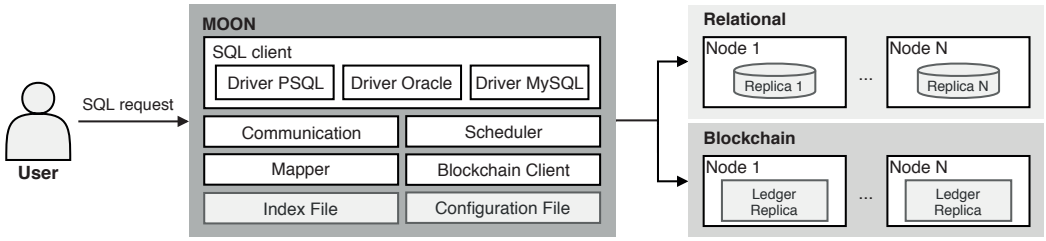


Figure 2. Architecture of MOON

In addition to the modules, MOON uses two files: the Configuration File and the Index File. The first contains the information needed to execute the modules, such as the addresses of the nodes, the adopted DBMS and Blockchain Infrastructure in use. The other file addresses a dissimilarity that exists between Blockchain and RDB. Generally, data has hash identifiers to be found in Blockchain. On the other hand, in RDB, other ways of identifying data are used, such as numerical data and text. Thus, the index file is used to match the identifier chosen by the user, via SQL, with the identifier used in Blockchain. If this match is not made to find specific data, it will probably be necessary to check all Blockchain data records.

Figure 3 shows the step-by-step of retrieving the data in MOON, from a SQL query, with the use of indexes, in a Blockchain. This figure is complementary to Figure 1 and is focused on what goes on inside MOON. In steps 1 through 3, the client query is sent to the Scheduler and is then routed to the Mapper. In step 4, the Mapper Module accesses the Index File to form a list with the identifiers of the data, in Blockchain, requested by the SQL query. After that, in Step 5, the Mapper requests this list of data to the Blockchain Client, which is responsible for retrieving them, this being Step 6. Then, the Blockchain Client returns the requested data to the Mapper, which organizes this data into the format of relation. Finally, after passing through the Scheduler Module, the table with the query response is returned by the Communication Module to the Client, it is in arrow 4 of Figure 1.

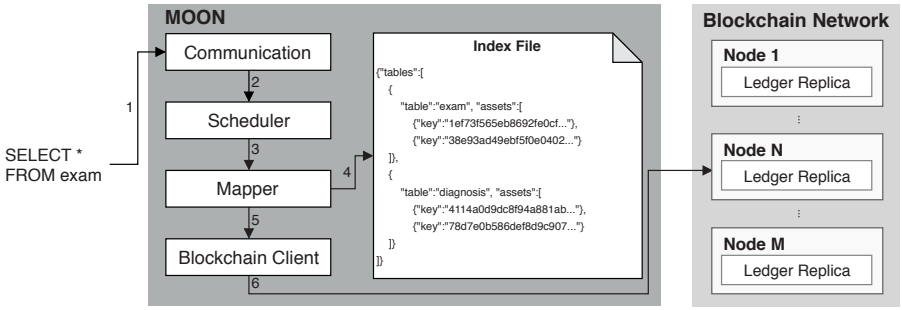


Figure 3. MOON Step-by-Step on Data Queries that are in Blockchain

4. Related Work

The following search criteria were used to collect some related works: (i) Works that use Blockchain or related concepts; and (ii) Works that use RDBs or their properties. The search for papers from 2014 to 2019 was carried out in the following repositories: ACM Digital Library, IEEE Xplore Digital Library, ScienceDirect, Proceedings of the Brazilian Symposium on Databases and Journal of Information and Data Management.

Hearn (2016) describes the R3 Corda, a permissioned Blockchain to the financial sector, which stores data in RDB. The objectives of this approach are: (i) to provide Blockchain properties such as immutability; (ii) to handle complex transactions, providing joins and analytics using SQL; and (iii) to restrict access to transaction data by the use of need-to-know protocols, which share information with only those involved in transactions, not with all nodes. The mapping done by R3 Corda is Object-Relational and uses Java Persistence API (JPA) annotations, which gives it dependent on this technology. Finally, as previously mentioned, this approach does not support applications of different purposes, because it is focused on financial contexts.

Bartolleti et. al (2017) proposed a framework to perform analytics on Blockchain data of cryptocurrency context, supporting Bitcoin and Ethereum. The approach is a Scala library that provides integration among Blockchain data and data from external sources, such as exchange rates between a cryptocurrency and dollar. The data is entered into a database, SQL or NoSQL, and is then used as a view. Next, analytical queries can be made using the communication interface provided by that database. Therefore, the research does not discuss how to manage data that is stored in different technologies since this is not the focus. However, the need to provide complex queries to Blockchains is addressed.

Bragagnolo et. al (2018) have developed the Ethereum Query Language (EQL), which allows users to retrieve information from a Blockchain Ethereum using queries similar to SQL. The developed language provides rich syntax, which supports specifying data elements to search for information spread across multiple records. Originally, to search for data in Ethereum, it is necessary to access the blocks using a unique identifier or to search multiple blocks sequentially to find the desired data. Thus, the research is based on a relevant aspect of the relational model but does not present hybridization characteristics between the Relational model and Blockchain. However, there is a proposed future work correlated with MOON, which is to use EQL to get data from different sources.

Muzammal, Qu, and Nasrulin (2019) proposed the ChainSQL, an approach that integrates Blockchain and RDB. It is decentralized, distributed and auditable, which are Blockchain characteristics. Besides, it provides fast query processing, typical of RDB. It because, according to the work, the blockchain properties mentioned above are relevant, but this technology does not make it possible to perform quick and complex queries. In ChainSQL, Blockchain stores transactions and RDB stores actual data, which makes it a database system that has Blockchain-based logging. Thus, this research does not aim the partitioning of data between Blockchain and Database, focusing on the synchronization of data and the transaction log of this data.

5. Current Status and Next Steps

The research was designed to be conducted in four stages and is currently at the beginning of the third stage. The first one was already concluded and consists of a literature review for the acquisition of the necessary theoretical basis: concepts of RDB and Blockchain, distributed data management and distributed concurrency control. The second stage comprised the search for related works carried out to identify the research opportunities in the area. The third step is to implement the proposed solution. The modules of Figure 2 and the algorithms to be used, such as those related to mapping the Relational model to Blockchain, will be implemented. The DBMS initially chosen is PostgreSQL, since it is free and widely used. The Blockchain infrastructure chosen is BigchainDB, since it has a more general purpose of use, accepting plain text records [BigchainDB GmbH 2018]. Finally, in the fourth step, the validation of the MOON will be carried out through a case study with real data. Currently, two datasets were obtained, one from the health context and the other from a juridical basis.

6. Submissions and Publications

A publication was held to present some research opportunities and challenges in Blockchain and distributed data management [Moreira et al. 2019]. In this research that was published, one of the opportunities identified is the one proposed by MOON. Then, after completing step four of this research, probably in December, an article for the Journal of Information and Data Management should be submitted.

7. Final Considerations

This research proposes the MOON, an approach to managing data in RDB and Blockchain, in a disjoint way. In the current state-of-the-art, it was verified that few

articles have a strong relation to the MOON. Despite the above mentioned, Bragagnolo et. al (2018) proposed as future work the use of a language to manage both data in Blockchain and in other sources, which is the MOON proposal. Currently, the project is at the beginning of the implementation phase. After this step, a case study will be carried out to validate or reject the research hypothesis.

Acknowledgment: This study was financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. This research was partially supported by LSBD/UFC - Brazil.

References

Bartoletti, M., Lande, S., Pompianu, L., and Bracciali, A. (2017). A general framework for blockchain analytics. In *Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*, SERIAL '17, pages 7:1–7:6, NY, USA. ACM.

BigchainDB GmbH (2018). BigchainDB 2.0: The Blockchain Database. `https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf`. Online; accessed 17 June 2019.

Bragagnolo, S., Rocha, H., Denker, M., and Ducasse, S. (2018). Ethereum query language. In *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, WETSEB '18, pages 1–8, NY, USA. ACM.

Chowdhury, M. J. M., Colman, A., Kabir, M. A., Han, J., and Sarda, P. (2018). Blockchain versus database: A critical analysis. In *2018 17th TrustCom / 12th BigDataSE*, pages 1348–1353. IEEE.

Greve, F., Sampaio, L., Abijaude, J., Coutinho, A., Ítalo Valcy, and Queiroz, S. (2018). *Blockchain e a Revolução do Consenso sob Demanda*, chapter 5, pages 1–52. SBC.

Hearn, M. (2016). Corda: A distributed ledger. *Corda Technical White Paper*.

Maenhaut, P., Moens, H., Ongenae, V., and De Turck, F. (2015). Design and evaluation of a hierarchical multi-tenant data management framework for cloud applications. In *2015 IFIP/IEEE IM*, pages 1208–1213.

Moreira, L. O., Marinho, C. S. S., Neto, M. M., Coutinho, E. F., de Souza, J. N., and Machado, J. C. (2019). Oportunidades de pesquisa para o uso de infraestruturas blockchain na gestão de dados distribuídos. *Revista Sistemas e Mídias Digitais*, 4(1).

Muzammal, M., Qu, Q., and Nasrulin, B. (2019). Renovating blockchain with distributed databases: An open source system. *Future Generation Computer Systems*, 90:105 – 117.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Yaga, D. and Mell, P. (2018). *NISTIR 8202: Blockchain Technology Overview*, chapter 1, pages 1–68. National Institute of Standards and Technology.

Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE International Congress on Big Data*, pages 557–564.

Avaliação de Confiabilidade das Viagens de Ônibus com base na Conformidade entre Dados de GPS e GTFS

Aluna: Andreza Raquel Monteiro de Queiroz

E-mail: andrezaraquel@copin.ufcg.edu.br

Orientador: Carlos Eduardo Santos Pires

E-mail: cesp@dsc.ufcg.edu.br

Universidade Federal de Campina Grande - UFCG
Programa de Pós-Graduação em Ciência da Computação

Nível: Mestrado

Mês e Ano de Ingresso: março/2018

Mês e Ano Previstos para Defesa: fevereiro/2020

Etapas Concluídas: Créditos em Disciplinas, Exame de Qualificação, Definição do Problema, Especificação da solução, Referencial Bibliográfico Inicial e Parte da Análise dos Dados.

Etapas Futuras: Finalização da Especificação da solução, Implementação, Realização de Experimentos e Escrita da Dissertação.

Resumo. *O GTFS (General Transit Feed Specification) é um padrão para disponibilização de serviços programados que deveriam ser fielmente seguidos pelos ônibus. Porém, na prática, ocorrem dois problemas: i) por serem estáticos, os dados de GTFS ficam desatualizados com frequência e ii) há discrepâncias entre dados de GPS de ônibus e de GTFS. Esses problemas prejudicam diretamente a confiabilidade das viagens dos ônibus. Nesse contexto, esta pesquisa visa avaliar a confiabilidade dessas viagens, com base na análise de conformidade entre os dados de GPS e GTFS. Como resultados parciais, inconsistências encontradas na análise foram enumeradas e as trajetórias de ônibus foram classificadas em relação à conformidade com as rotas do GTFS.*

1. Introdução

A confiabilidade das viagens é um dos fatores que mais influenciam a escolha do modo de locomoção dos passageiros [Ma et al. 2016]. Fontes indicam que tais passageiros estão progressivamente migrando para o transporte privado por não confiarem no sistema de transporte público. Somente em 2017, a demanda de passageiros caiu 9,5%, segundo a NTU (Associação Nacional das Empresas de Transportes Urbanos) [NTU 2017]. Na cidade de Campina Grande, por exemplo, registrou-se uma queda de 11% dos usuários de ônibus no primeiro semestre de 2018 em relação ao mesmo período de 2017. Esse número representa um prejuízo de cerca de 4,9 milhões de reais para as empresas[1]

No que diz respeito a viagens de ônibus, a falta de confiabilidade está diretamente relacionada ao não cumprimento das rotas e dos horários programados. Em geral, os ônibus não cumprem uma rota por dois motivos: i) por estarem lotados - nesse caso, o motorista opta por mudar a rota ignorando algumas paradas de forma a evitar embarques de passageiros; e/ou ii) por estarem atrasados em relação ao horário programado - nesse caso, o motorista deseja terminar a viagem rapidamente, optando por um caminho mais curto. Em relação ao não cumprimento dos horários programados, isso pode acontecer porque: i) houve alteração na demanda de passageiros e uma consequente diminuição/aumento no tempo de embarque/desembarque dos passageiros; e/ou ii) ocorreram eventos estocásticos no trânsito, como acidentes e engarrafamentos.

Os dados de rotas e horários programados das viagens de ônibus são usualmente disponibilizados no GTFS[2] (*General Transit Feed Specification*). Basicamente, o GTFS é um padrão para disponibilização de dados de transporte público, composto por alguns arquivos. Cada arquivo de GTFS armazena informações de um serviço programado e oferecido pelas empresas de transporte. Além dos arquivos de rotas e de horários programados para cada viagem realizada durante os dias, o GTFS contém arquivos que indicam o prazo de validade das informações nele contidas, preço da passagem, entre outros.

Ao analisar as informações do GTFS em relação ao que acontece na pratica (i.e., em relação aos serviços que os ônibus estão prestando), percebe-se que existe uma falta de conformidade entre os dados de GTFS e de GPS emitidos pelos ônibus. Os dados de GPS consistem em registros contendo *timestamp*, identificador do veículo, rótulo da rota, latitude e longitude. A falta de conformidade ocorre, por exemplo, quando o rótulo da rota indicado nos dados de GPS é diferente da rota que o ônibus está realmente percorrendo. Isso pode acontecer porque houve alguma mudança nos serviços que não foi refletida no GTFS, provocando a desatualização das informações nele contidas, mesmo que seu prazo de validade ainda não tenha expirado. Um problema ainda mais crítico, do ponto de vista da análise dos dados, ocorre quando o rótulo da rota não está presente nos dados de GPS, provavelmente devido a falhas do dispositivo que emite esses dados [Raymond and Imamichi 2016].

Nesse contexto, esta pesquisa visa avaliar a confiabilidade das viagens de ônibus, analisando a conformidade dos dados de GPS com os dados de GTFS. Na análise, foram encontradas e classificadas inconsistências entre esses dados. No fim, um valor de confi-

[1]Transporte público de Campina Grande perde usuários em 2018: https://g1.globo.com/pb/paraiba/noticia/2018/11/08/empresarios-querem-cobrar-por-viagens-de-onibus-a-partir-da-integracao-em-campina-grande.ghtml

[2]https://developers.google.com/transit/gtfs/reference/

abilidade será atribuído a cada viagem, no intuito de compor um nível de confiabilidade para cada rota e para o serviço de transporte por ônibus da cidade como um todo.

2. Trabalhos Relacionados

Estudos sobre a confiabilidade do sistema de transporte público são vastamente difundidos na literatura, tanto do ponto de vista do passageiro quanto das empresas de transporte. Uma vez que nessa pesquisa o foco é a perspectiva do passageiro, alguns indicadores de confiabilidade podem ser destacados: aderência aos horários programados; tempo de viagem; disponibilidade de informações sobre rotas e serviços; e frequência com que os ônibus passam nas paradas.

Nesse contexto, em [Chen et al. 2009] a confiabilidade das viagens de ônibus é analisada de acordo com os seguintes indicadores: desvio dos horários programados de chegada dos ônibus nas paradas; desvios das paradas; e uniformidade entre tempos de chegadas dos ônibus nas paradas. Para cada um desses indicadores, uma métrica é proposta. Entretanto, apesar de comparar dados reais (GPS dos ônibus) com dados programados, os autores não consideram a falta conformidade que pode ocorrer entre esses dados.

Em um trabalho mais recente, os autores de [Pi et al. 2018] usam dados APC (*Automatic Passenger Counting*) e de GPS para avaliar a confiabilidade das viagens. A partir desses dados, os autores criaram indicadores de confiabilidade relacionados ao tempo de espera do passageiro, à frequência com que os ônibus deixam de passar em paradas, à ocorrência de aglomerados de ônibus, ao tempo de viagem, ao cumprimento de horários programados e à lotação dos ônibus. Apesar de agregar informações importantes para o desenvolvimento de métricas de confiabilidade, os dados APC são de difícil acesso por apresentarem restrições de confidencialidade.

Os autores de [Rajabi-Bahaabadi et al. 2019] propõem a análise de confiabilidade das viagens de ônibus, utilizando novas métricas. Tais métricas consideram as atitudes dos passageiros em relação ao risco de chegar ao destino fora do horário. A intuição utilizada pelos autores é que um passageiro que vai ao parque de diversões em um dia de lazer percebe e sofre menos as consequências de um atraso do que um passageiro que está indo em direção ao trabalho, por exemplo. Entretanto, classificar as viagens com base nessa intuição é uma forma de camuflar a falta de confiabilidade do sistema de transporte analisado. Isso porque os atrasos continuam ocorrendo, mesmo que apenas alguns passageiros sejam afetados por eles.

No que diz respeito aos problemas de falta de conformidade entre dados de GPS e GTFS descritos na Seção 1, existem dois trabalhos principais na literatura relacionada. O primeiro trabalho está relacionado ao problema de desatualização do GTFS. Os autores de [Wessel et al. 2017] desenvolveram um método para melhorar a precisão dos dados GTFS usando dados de GPS de ônibus em tempo real. Basicamente, o método atualiza os dados do GTFS cada vez que uma mudança significativa na trajetória (conjunto de registros de GPS, ordenado pelo timestamp, contendo o mesmo identificador de veículo) de um ônibus é detectada. No entanto, os autores não lidam com os problemas de GPS sem rótulo de rota ou indicando uma rota que não é a realizada pelo ônibus. Isso pode gerar atualizações erradas no GTFS.

O segundo trabalho endereça o problema da ausência do rótulo da rota nos dados de GPS e foi proposto por [Raymond and Imamichi 2016]. Nele, os autores mostraram

que a semelhança de cosseno é um método eficaz para determinar as rotas seguidas pelos ônibus. Em seus experimentos, usando dados do Rio de Janeiro, eles compararam os resultados gerados usando a semelhança do cosseno com o rótulo de rota presente nos dados de GPS (dados rotulados de GPS foram utilizados para realizar a validação do método). No entanto, os autores não consideraram o problema de falta de conformidade entre o rótulo de rota do GPS e as rotas do GTFS, confiando inteiramente no rótulo presente no GPS como sendo o gabarito.

Nenhum dos trabalhos que avaliam a confiabilidade de viagens descritos endereça problemas de conformidade entre os dados. Por isso, um dos objetivos dessa pesquisa consiste em avaliar a possibilidade de extensão das métricas de confiabilidade já existentes, considerando também um novo indicador relacionado à conformidade entre os dados de GPS e GTFS.

3. Metodologia

Utilizando essencialmente a modelagem como método científico, esta pesquisa divide-se basicamente em três fases. A primeira fase é dedicada ao levantamento do estado da arte que está sendo feito desde o início e continuará sendo realizado até o final da pesquisa. Nessa fase, foram elencados os trabalhos relacionados ao processamento de dados de transporte público e à confiabilidade de viagens de ônibus. A segunda fase diz respeito à análise de conformidade entre dados de GPS e GTFS, e está em andamento. Na terceira fase, será possível avaliar a confiabilidade das viagens a partir de um estudo das métricas do estado da arte.

Nessa pesquisa estão sendo utilizados conjuntos de dados de três cidades brasileiras: Cidade A (nome da cidade omitido devido a requisitos de privacidade relacionados ao uso de seus dados de GPS), Curitiba e Rio de Janeiro. Esses conjuntos de dados estão resumidos na Tabela 1. Os dados de GTFS e GPS de Curitiba[3] foram obtidos na página da URBS (Urbanização de Curitiba S / A), agência que administra o transporte público em Curitiba. Os dados da Cidade A foram disponibilizados pela agência que administra o transporte público na cidade. Os dados de GTFS e GPS do Rio de Janeiro foram fornecidos pelos autores de [Raymond and Imamichi 2016].

Tabela 1. Sumário dos dados.

	Intervalo dos dados de GPS	# Trajetórias	# Rotas do GTFS
Cidade A	03/12/2018 a 07/12/2018	247	250
Curitiba	27/08/2017 a 31/08/2017	697	235
Rio de Janeiro	15/02/2016 a 17/02/2016	5.376	375

4. Solução Proposta

Esta seção descreve o processo de avaliação da confiabilidade das viagens de ônibus. A Figura 1 apresenta o fluxograma das etapas do processo. A primeira etapa consiste em realizar a análise de conformidade entre dados de GPS e de GTFS. Essa etapa é necessária porque o rótulo de rota presente nos dados de GPS pode não corresponder à rota seguida

[3]http://dadosabertos.c3sl.ufpr.br/curitibaurbs/, http://transporteservico.urbs.curitiba.pr.gov.br/index.php

pelo ônibus. Na segunda etapa, é feita a segmentação de trajetórias em viagens porque uma trajetória contém dados de GPS emitidos por um ônibus durante vários dias e engloba várias viagens (subconjunto da trajetória que cumpre a rota apenas uma vez). Por fim, a última etapa diz respeito à avaliação de confiabilidade que será realizada em cada viagem obtida da etapa anterior.

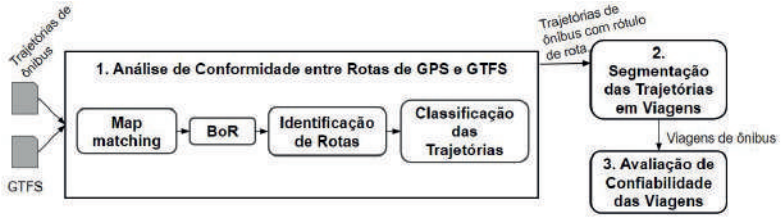


Figura 1. Fluxograma do processo para avaliação de confiabilidade das viagens de ônibus.

Para analisar a conformidade entre os dados de GPS e os dados das rotas contidas no GTFS (Passo 1 da Figura 1), foi necessário seguir três passos baseados na abordagem proposta por [Raymond and Imamichi 2016]. O primeiro passo consistiu em aplicar *map matching* (MM) nas trajetórias de ônibus e rotas do GTFS. MM é um processo que integra um mapa a dados de posicionamento ruidosos para obter posições acuradas. Nessa pesquisa, o algoritmo de MM empregado foi o GraphHopper[4]. Basicamente, esse algoritmo segue a abordagem do estado-da-arte descrita em [Newson and Krumm 2009], a qual é baseada em cadeias de Markov (*Hidden Markov Model* ou HMM). Métodos de MM baseados em HMM são conhecidos por serem robustos em lidar com dados de GPS que podem conter erros, bem como intervalos entre medições longos e irregulares [Kubicka et al. 2018]. A última versão do *Open Street Map*[5] foi usada como mapa. Em relação aos dados do Rio de Janeiro, foi usado o resultado da fase de MM disponibilizado pelos autores de [Raymond and Imamichi 2016].

No passo seguinte, as saídas da fase de MM (i.e., as posições acuradas) foram transformadas em vetores. Nesse passo, foram gerados vetores *bag-of-roads* (BoR) para representar as trajetórias de ônibus e rotas do GTFS. Os vetores BoR mantêm a mesma dimensionalidade. Dessa forma, é possível realizar comparações entre eles usando métricas clássicas de similaridade, como a similaridade do cosseno. Vetores BoR funcionam analogamente aos vetores *bag-of-words* na classificação de documentos. No entanto, em vez de contar palavras, cada célula de um vetor BoR representa um segmento de rua e armazena a frequência que um ônibus ou uma rota atravessa aquele segmento de rua.

O terceiro passo consistiu em identificar a rota GTFS seguida por cada ônibus. Para isso, a similaridade do cosseno entre pares de vetores BoR foi calculada. Cada par contém um vetor BoR representando trajetórias de ônibus e um vetor BoR representando rotas GTFS. O resultado dessa etapa é uma lista de trajetórias de ônibus, cada uma delas associada à sua rota mais similar. Em alguns casos, a rota mais similar encontrada não coincide com a rota indicada nos dados de GPS, confirmando a existência de inconsistências na rota dos dados de GPS, como mencionado na Seção 1.

[4]Código do *GraphHopper* disponível em: https://github.com/graphhopper/map-matching
[5]http://download.geofabrik.de/

Ao final desses três passos, uma classificação de trajetórias é proposta com base no valor de similaridade do cosseno com a rota mais similar. Quanto maior o valor de similaridade, maior a conformidade entre as trajetórias e as rotas do GTFS. As classes são as seguintes: i) há alto desvio de rota, ou não há rota correspondente no GTFS - quando a similaridade é baixa (menor que um dado limiar); ii) o ônibus segue a rota rotulada nos dados de GPS consistentemente - quando o rótulo de rota do GPS corresponde à rota encontrada utilizando o cosseno e a similaridade é alta (maior que o dado limiar); e iii) o ônibus segue outra rota consistentemente - quando o rótulo de rota do GPS não corresponde à rota encontrada utilizando o cosseno mas a similaridade é alta.

5. Resultados Parciais

Como resultado da análise de conformidade descrita na Seção 4, foi possível identificar/confirmar a ocorrência das seguintes inconsistências: i) o rótulo da rota dos dados de GPS é diferente da rota que o ônibus está percorrendo; ii) o ônibus desvia parcialmente da rota; iii) o ônibus percorre mais de uma rota, embora o rótulo do GPS indique apenas uma rota; iv) o ônibus não percorre nenhuma das rotas presentes no GTFS.

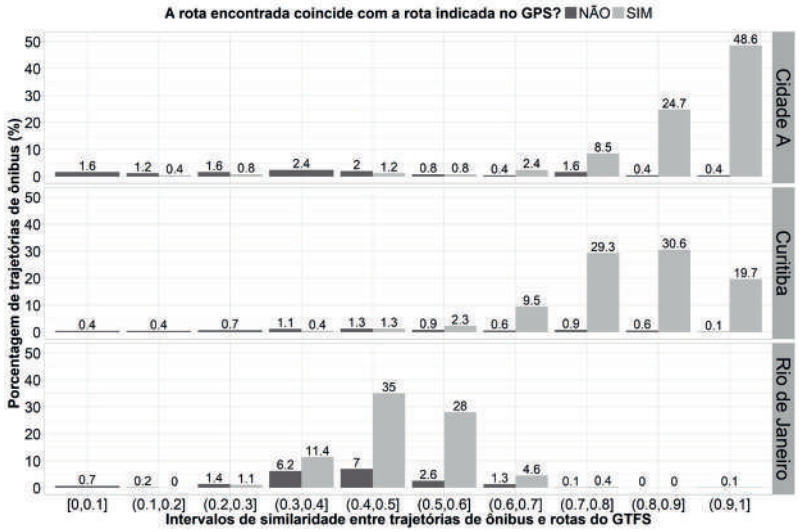


Figura 2. Porcentagem de trajetórias, por intervalo de similaridade, cujas rotas indicadas nos dados de GPS coincidem (ou não) com a rota encontrada.

A Figura 2 apresenta a porcentagem de trajetórias de ônibus cujas rotas do GTFS mais similares encontradas pelo método cosseno correspondem (ou não) às rotas indicadas nos dados de GPS, por intervalo de similaridade em cada cidade. O eixo horizontal representa os intervalos de similaridade, enquanto o eixo vertical representa a porcentagem de trajetórias de ônibus. As barras cinza escuro referem-se às trajetórias de ônibus cujas rotas indicadas nos dados de GPS são diferentes das rotas mais similares de acordo com a distância do cosseno. Por sua vez, as barras cinza claro referem-se às trajetórias de ônibus cujas rotas indicadas nos dados de GPS correspondem às rotas mais similares encontradas utilizando a distância do cosseno. Percebe-se que a maioria dos ônibus da Cidade A e de Curitiba percorre a rota indicada no GPS, enquanto 18.2% (Cidade A) e 20.4% (Curitiba) das trajetórias apresentam similaridade menor que 0.7. Por outro lado, o Rio de Janeiro apresenta um cenário diferente, uma vez que as trajetórias de ônibus estão concentradas entre os valores de similaridade 0.3 e 0.6.

6. Considerações Parciais e Próximos Passos

A pesquisa descrita nesse trabalho propõe uma análise de conformidade de dados com a finalidade de avaliar a confiabilidade das viagens de ônibus em relação aos dados programados contidos no GTFS. Atualmente, a pesquisa encontra-se na finalização da etapa de *Análise de Conformidade entre Rotas de GPS e GTFS* (Figura 1). Diversas inconsistências entre os dados de GPS e GTFS foram encontradas e classificadas. Em etapas futuras, será calculada a confiabilidade das viagens dos ônibus. Para isso, será realizado um estudo das métricas de confiabilidade de viagens existentes. A possibilidade de proposição de uma nova métrica que endereça a falta de conformidade entre dados de GPS e GTFS será considerada. Dessa forma, será possível classificar as viagens de todas as rotas existentes nas cidades de acordo com seu nível de confiabilidade do ponto de vista do usuário.

Agradecimentos

Esta pesquisa foi parcialmente financiada pelo INES 2.0, concessão da FACEPE APQ-0399-1.03/17, concessão da CAPES 88887.136410/2017-00 e concessão do CNPq 465614/2014-0.

Referências

Chen, X., Yu, L., Zhang, Y., and Guo, J. (2009). Analyzing urban bus service reliability at the stop, route, and network levels. *Transportation research part A: policy and practice*, 43(8):722–734.

Kubicka, M., Cela, A., Mounier, H., and Niculescu, S.-I. (2018). Comparative study and application-oriented classification of vehicular map-matching methods. *IEEE Intelligent Transportation Systems Magazine*, 10(2):150–166.

Ma, Z., Ferreira, L., Mesbah, M., and Zhu, S. (2016). Modeling distributions of travel time variability for bus operations. *Journal of Advanced Transportation*, 50(1):6–24.

Newson, P. and Krumm, J. (2009). Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343. ACM.

NTU (2017). Demanda por passageiros de transporte público cai em 2017. Disponível em: https://www.ntu.org.br/novo/upload/Publicacao/Pub636687203994198126.pdf. Acesso em 01 de julho de 2019.

Pi, X., Egge, M., Whitmore, J., Silbermann, A., and Qian, Z. S. (2018). Understanding transit system performance using avl-apc data: An analytics platform with case studies for the pittsburgh region. *Journal of Public Transportation*, 21(2):2.

Rajabi-Bahaabadi, M., Shariat-Mohaymany, A., and Yang, S. (2019). Travel time reliability measures accommodating scheduling preferences of travelers. *Transportation Research Record*, 2673(4):708–721.

Raymond, R. and Imamichi, T. (2016). Bus trajectory identification by map-matching. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1618–1623. IEEE.

Wessel, N., Allen, J., and Farber, S. (2017). Constructing a routable retrospective transit timetable from a real-time vehicle location feed and gtfs. *Journal of Transport Geography*, 62:92–97.

Main memory databases instant recovery

Arlino Magalhães[1,2], José Maria Monteiro[1], Angelo Brayner[1]

[1]Universidade Federal do Ceará
Fortaleza – CE – Brasil

[2]Universidade Federal do Piaui
Teresina – PI – Brasil

arlino@ufpi.edu.br, {monteiro,brayner}@dc.ufc.br

Abstract. *The in-memory databases recover offline, meaning that database only becomes available for new transactions after the full recovery process. Systems can keep database replicas for high availability. However, hardware improvements are not immune to some sources of failures that can cause multiple and shared failures. Thus, software techniques are required to prevent failures and repair failed systems as quickly as possible. This work proposes an instant recovery approach for in-memory databases, i.e., the system can process transactions immediately after the failure. Our approach can redo the data into memory incrementally and on-demand to meet transactions as soon as they arrive.*

Resumo. *Os bancos em memória recuperam-se offline, significando que o banco de dados só torna-se disponível para novas transações após o completo processo de recuperação. Sistemas podem manter replicas do banco de dados para alta disponibilidade. Entretanto, réplicas não estão imunes a fontes de falhas, que podem causar falhas múltiplas e compartilhadas. Por esse motivo, técnicas de software são necessárias para evitar falhas e para reparar sistemas que falharam o mais rápido possível. Este trabalho propões uma abordagem de recuperação instantânea para bancos de dados em memória, ou seja, o sistema pode processar transações imediatamente após a falha. Nossa abordagem pode refazer os dados na memória incrementalmente e sob-demanda para atender a novas transações assim que eles chegam.*

1. General information

- **Title:** Main memory databases instant recovery
- **Level:** doctorate degree
- **Student full name:** Arlino Henrique Magalhães de Araújo - arlino@ufpi.edu.br
- **Advisor:** José Maria Monteiro - monteiro@dc.ufc.br
- **Co-advisor:** Angelo Brayner - brayner@dc.ufc.br
- **University / Program:** Federal University of Ceara / Master's and Doctoral Program in Computer Science
- **Date of entry / Date of defense :** March 2016 / December 2020
- **Completed steps and Completion period:**
  - disciplines - 2018-1, and
  - qualification - 2018-2.

2. Introduction and Motivation

Main Memory Database - MMDB (or In-Memory Database - IMDB) reduces the disk I/O bottleneck and thus it can achieve very high-speed access. However, the primary copy of the database in volatile main memory is more vulnerable to failures than conventional disk-resident databases. Recovery activities (logging, checkpoint and reloading) are used to restore an MMDB to the most consistent state after a system crash has occurred [Malviya et al. 2014, Mohan et al. 1992].

After a system crash, the lost in-memory database can be recovered by reloading a archive checkpoint (backup copy) residing on secondary memory into main memory and replaying log data also from secondary memory. Efficient reload and replay schemes are essential to ensure that the expectations of high performance database systems are met. However, as the database is not available for transactions during the recovery process, the system performance severely degrades [Gruenwald and Eich 1994].

This paper proposes an instant recovery approach for OLTP in-memory databases, i.e., the transactions can run immediately after the failure. During recovery, our approach allows the system to load first most frequently accessed data. Thus the system can meet new transactions more efficiently. Moreover, the approach can retrieve tuples on-demand to meet transactions whose data has not yet been loaded into memory. In the remainder of this paper, Section 3 provides a background, Section 4 exposes the related work, Section 5 describes the problem definition, Section 6 describes our proposed solution, and Section 10 concludes.

3. Background

3.1. Main memory database recovery

Most IMDBs perform a logical redo logging to reduce the amount of data written to secondary storage. The log records contain only after images of modified tuples with descriptions of higher-level operations, such as to insert a record in a table. Commit processing attempts to group multiple log records into one large I/O (group commit). Periodically, the system asynchronously produces a consistent checkpoint archive (snapshot) on disk in order to reduce the recovery time and free up log space. Snapshot is equivalent to a materialized database state at a given time. MMDBs restore by loading the last valid checkpoint and then replay a logical log forward from checkpoint timestamp [Zheng et al. 2014, Wu et al. 2017, Faerber et al. 2017].

3.2. Hot and cold data

In OLTP workloads, some records are accessed frequently (hot) and others infrequently accessed (cold). The system performance depends on the frequency of hot records residing in memory. Thus, cold records should be moved from main memory to secondary storage to free up space for hot records. Anti-caching [DeBrabant et al. 2013] and Siberia [Eldawy et al. 2014] are examples of techniques to manage hot and cold data.

4. Related work

Hekaton [Diaconu et al. 2013], VoltDB [Stonebraker and Weisberg 2013], H-Store [Kallman et al. 2008], HyPer [Funke et al. 2014], SAP HANA [Färber et al. 2012] and

SiloR [Zheng et al. 2014] are examples of modern in-memory databases systems that perform checkpoint and logging activities to recover a database after a failure. PACMAN [Wu et al. 2017] and Adaptive Logging [Yao et al. 2016] produce a dependency graph between transactions performed to identify opportunities to recover the database parallel. In those systems, the database permits new transaction only after the full recovery is completed and they can keeps database replicas for high availability.

Sauer et al. [Sauer et al. 2017, Sauer et al. 2018] present a technique to restore a disk-resident database from a media failure. This technique indexes the log records. After a failure, the recovery process loads pages from a backup device and log records pertaining to these pages are probed in a log index. New transactions can perform during recovery as soon as their necessary pages are restored.

5. Problem definition

The MMDB recovery process is performed offline, meaning that the database and its applications become available for new transactions only after the full recovery process is completed. However, systems can maintain database replicas for high availability. With the advent of high-availability infrastructure, recovery speed has become secondary in importance to run-time performance for most of the in-memory database systems [Wu et al. 2017, Faerber et al. 2017, Malviya et al. 2014]. However, replicas are not immune to human errors and unpredictable defects in software and firmware that can be source of failures and can cause multiple and shared failures. Thus software techniques are necessary to avoid failures and repair failed systems as quickly as possible [Sauer et al. 2017, Sauer et al. 2018].

6. A new instant recovery approach

6.1. The proposed architecture

In Figure 1, we propose an architecture to implement an in-memory database instant recovery approach. The main components of our architecture follows below:

- **Logger**: records transaction update actions in a log archive on secondary memory.
- **Indexer**: indexes the log records to a sequential file.
- **Access logger**: records transaction access information for tuples.
- **Classifier**: identifies hot/cold tuple data and records this information.
- **Checkpoint**: reads hot/cold data and memory data to create a snapshot on disk. The data in the snapshot is sorted according to the hot and cold data classification.
- **Restorer**: after a failure, load the snapshot data. For each data portion loaded, a log retry is done using the indexed log.
- **Scheduler**: requests tuples not yet loaded into memory for Restorer.

In the following discussion, the numbers in parentheses refer to the numbered steps in Figure 1. During normal transactions processing, the Logger component records transactions update actions on a log achieve (1). The Indexer component monitors entries in log (2) and stores them in a index structure (3). The log indexing can be done asynchronous to transactions commit, i.e., the transaction doesn't need to wait for the indexing to confirm its written since the written on sequential log already ensures the durability. Section 6.2 explain in more details our indexed log.

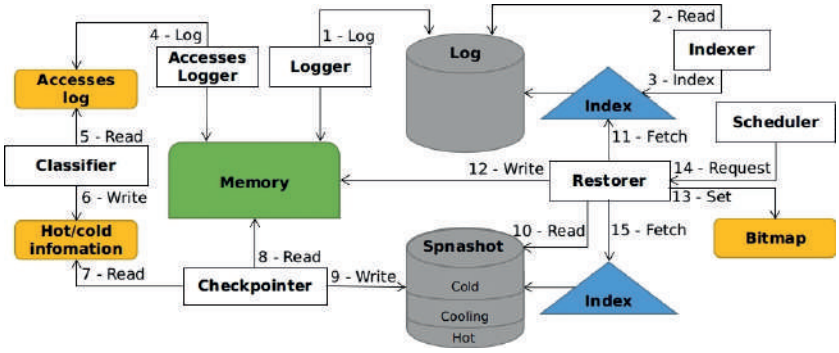


Figure 1. Architecture for in-memory database instant recovery.

The Accesses Logger component records accesses to tuples from transactions in an Accesses Log component (4). Periodically, the Classifier reads the data from the Accesses Log (5), identifies hot/cold data, and records this information (6). Also periodically, the Checkpoint reads the hot/cold data information (7); reads data into memory (8); and creates a snapshot according to hot and cold data classifying (9). Indexes can be created in the snapshot to improve fetches.

After a system failure, the restore manager starts the recovery loading tuples on the last snapshot (10). The Restorer loads portions of tuples (segments). The segments are loaded from hottest to coldest tuples. After a segment is loaded, a log replay must be done for each tuple of the segment. Thus Restorer fetches log records of these tuples in the index structure (11). Then those tuples will written into memory in the last consistent state before the failure (12). Once in memory, a segment is set as restored in bitmap (13).

The system can perform transactions during the recovery process. Since tuples were stored according to the classification hot and cold data, probably tuples required for a new transaction will be loaded first. However, if tuples necessary for a transaction have not yet been loaded into memory, the Scheduler must request these tuples to Restorer (14). Thus Restorer must load segments for those tuples on snapshot (15) and replay log records for the segments. When the segments are restored, they must be updated as restored in bitmap and the transaction can be executed.

6.2. Logging strategy

In the proposed approach, each transaction generates redo records that are kept in a thread-local. During commit, all log records created by the transaction are appended in a sequential log atomically. Periodically, sequential log records are stored asynchronously in the indexed log. After indexing the log records, these records in the sequential log can be discarded. The example of Figure 2 contrasts the traditional sequential log (a) with the indexed log (b). In this example, three transactions Tx1, Tx2, and Tx3 generate log records for three tuples Tp1, Tp2, and Tp3. Sequential log records are appended in the same order as their transaction updates are performed on the database. It's necessary a full scan on the log to redo a single tuple. On the other hand, only an index B-tree search can retrieve all records to redo a single tuple. In the indexed log, records are stored in a B-Tree by row ID. Each node points to a queue of transaction update records of a tuple.

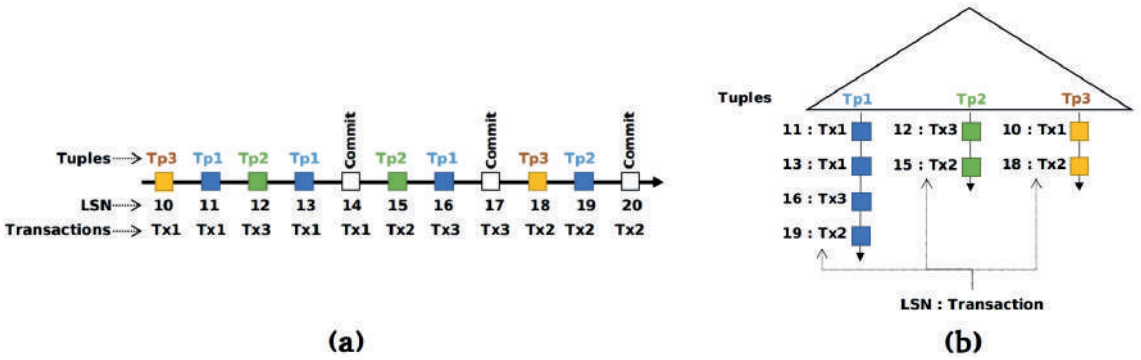


Figure 2. Sequential log (a) and indexed log (b).

6.3. Contributions

This work has the following contributions:

1. A generic approach for instant recovery of in-memory database. The transactions can perform during the database recovery leading the impression that the system was restored instantly.
2. A checkpoint approach based on the classification of hot and cold data. The checkpoint saves snapshots by sorting the data from the hottest to the coldest data.
3. An incremental recovery approach that loads data from hottest to coldest data. Hot data is loaded first, once it probably will be requested for new transactions.
4. An on-demand recovery approach that loads tuples as soon as a transaction arrives. If tuples needed by transaction have not yet been loaded into memory, the system can retrieves them on-demand.
5. An implementation of the proposed instant restore approach for main memory database.

7. Research Methodology

The research project consists of several activities, which are detailed below.

1. Bibliographic review about database recovery techniques (completed phase).
2. Selection of the most suitable techniques for in-memory databases recovery (completed phase).
3. Specification of the proposed techniques (completed phase).
4. Implementation and evaluation of the proposed techniques (developing phase).
5. Identification of optimization and improvements of techniques (unstarted phase).
6. Documentation of the system through technical report and scientific article (developing phase).
7. Thesis writing (unstarted phase).

8. Article submission planning

The implementation of the proposed architecture will be divided into modules. The research and results obtained in each module will be documented in articles that will be sent for publication in scientific conferences and journals as follows.

1. A survey article of main memory database recovery for an international journal in 2019.

2. Implementation of indexed log for in-memory database instant recovery for publication in national scientific conference in 2019.
3. Checkpoint implementation using hot and cold data classification for instant in-memory database recovery. The result should be published in an international scientific conference in 2020.
4. Implementation of possible improvements on system, such as a transaction-level logical logging, for example. The result should be published in an international scientific conference in 2020.

9. Preliminary evaluation of results obtained

The indexed log recovery approach proposed in this paper was implemented for physical log in a custom storage engine to evaluate the feasibility of indexes organization for log replay. Due space limitations, we could not present the results of the experiments in this paper. The result of experiments showed that our instant recovery approach improves the failure repair time and perceived mean time to recover the database. Our empirical analysis showed that instant recovery is able to effectively deliver tuples that new transactions need right after a failure. The experiments confirmed our expectation that a log indexing on transaction commit can degrade the transaction throughput. Thus, asynchronous log indexing to transaction commit is essential to our instant recovery approach and, consequently, to overall system performance.

10. Conclusions

This paper proposed am instant recovery approach for in-memory databases, allowing transactions to run together with the recovery process. Our checkpoint approach utilizes identifying hot/cold data to create snapshots according to a hot/cold data classification. This method permits, after a failure, to load incrementally the most frequently data accessed first, once these data probably will be requested for new transactions. In order to restore the data efficiently, our approach utilizes a indexed log to fetch tuples directly on log. Moreover, our approach can load data on-demand to meet transactions whose data has not yet been loaded into memory.

References

DeBrabant, J., Pavlo, A., Tu, S., Stonebraker, M., and Zdonik, S. (2013). Anti-caching: A new approach to database management system architecture. *Proceedings of the VLDB Endowment*, 6(14):1942–1953.

Diaconu, C., Freedman, C., Ismert, E., Larson, P.-A., Mittal, P., Stonecipher, R., Verma, N., and Zwilling, M. (2013). Hekaton: Sql server's memory-optimized oltp engine. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 1243–1254. ACM.

Eldawy, A., Levandoski, J., and Larson, P.-Å. (2014). Trekking through siberia: Managing cold data in a memory-optimized database. *Proceedings of the VLDB Endowment*, 7(11):931–942.

Faerber, F., Kemper, A., Larson, P.-Å., Levandoski, J., Neumann, T., Pavlo, A., et al. (2017). Main memory database systems. *Foundations and Trends® in Databases*, 8(1-2):1–130.

Färber, F., Cha, S. K., Primsch, J., Bornhövd, C., Sigg, S., and Lehner, W. (2012). Sap hana database: data management for modern business applications. *ACM Sigmod Record*, 40(4):45–51.

Funke, F., Kemper, A., Mühlbauer, T., Neumann, T., and Leis, V. (2014). Hyper beyond software: Exploiting modern hardware for main-memory database systems. *Datenbank-Spektrum*, 14(3):173–181.

Gruenwald, L. and Eich, M. H. (1994). Mmdb reload concerns. *Information sciences*, 76(1):151–176.

Kallman, R., Kimura, H., Natkins, J., Pavlo, A., Rasin, A., Zdonik, S., Jones, E. P., Madden, S., Stonebraker, M., Zhang, Y., et al. (2008). H-store: a high-performance, distributed main memory transaction processing system. *Proceedings of the VLDB Endowment*, 1(2):1496–1499.

Malviya, N., Weisberg, A., Madden, S., and Stonebraker, M. (2014). Rethinking main memory oltp recovery. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 604–615. IEEE.

Mohan, C., Haderle, D., Lindsay, B., Pirahesh, H., and Schwarz, P. (1992). Aries: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. *ACM Transactions on Database Systems (TODS)*, 17(1):94–162.

Sauer, C., Graefe, G., and Härder, T. (2017). Instant restore after a media failure. In *Advances in Databases and Information Systems*, pages 311–325. Springer.

Sauer, C., Graefe, G., and Härder, T. (2018). Fineline: log-structured transactional storage and recovery. *Proceedings of the VLDB Endowment*, 11(13):2249–2262.

Stonebraker, M. and Weisberg, A. (2013). The voltdb main memory dbms. *IEEE Data Eng. Bull.*, 36(2):21–27.

Wu, Y., Guo, W., Chan, C.-Y., and Tan, K.-L. (2017). Fast failure recovery for main-memory dbmss on multicores. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 267–281. ACM.

Yao, C., Agrawal, D., Chen, G., Ooi, B. C., and Wu, S. (2016). Adaptive logging: Optimizing logging and recovery costs in distributed in-memory databases. In *Proceedings of the 2016 International Conference on Management of Data*, pages 1119–1134. ACM.

Zheng, W., Tu, S., Kohler, E., and Liskov, B. (2014). Fast databases with fast durability and recovery through multicore parallelism. In *OSDI*, volume 14, pages 465–477.

Learning Individual Profiles behind Shared Accounts

Carolina Nery[1], **Renata Galante**[1], **Weverton Cordeiro**[1]

[1]Institute of Informatics - Federal University of Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brazil

{carolina.nery,galante,weverton.cordeiro}@inf.ufrgs.br

Level: Master Degree
Admission: March 2018
Expected conclusion: March 2020
Our M.Sc. program does not have Qualification Exam.

Abstract. *Various recommendation systems rely on users' account history (like visited/purchased/rated items) to predict which other items one could also be interested in. In practice, multiple individuals (e.g. family members or friends) may share a single account. For this reason, learning a single user profile from one's entire account history may lead to imprecise item suggestions. In this work, we propose tracking individual profiles behind shared accounts to better personalize item suggestions for the person currently in. In summary, we approach this problem by (*i*) identifying individual users' online sessions in a platform, and (*ii*) clustering them to learn the various users' profiles behind the (potentially) shared account.*

Keywords: Recommender System, Clustering, Item-item Similarity

Resumo. *Vários sistemas de recomendação dependem do histórico de contas dos usuários (como itens visitados/comprados/classificados) para prever quais outros itens eles podem ter interesse. Na prática, várias pessoas (por exemplo, membros da família ou amigos) podem compartilhar uma única conta. Por esse motivo, extrair um único perfil de usuário a patir do histórico de uma conta pode levar a imprecisões nas sugestões de itens. Neste trabalho, propõe-se a identificação de perfis individuais por trás de contas compartilhadas para melhor personalizar a recomendação de itens para a pessoa online no momento. Em resumo, o problema é resolvido (*i*) identificando as sessões online em uma plataforma, e (*ii*) clusterizando essas sessões para se identificar os perfis dos usuários por trás da conta (potencialmente) compartilhada.*

Palavras-chave: Sistemas de Recomendação, Clusterização, Similaridade entre Itens

1. Introduction and Problem Statement

Various entertainment, e-commerce, and media outlet companies, seeking to offer the best user experience possible, work hard on recommending items (like movies, products or news) their customers most likely will be interested in [Smith and Linden 2017]. For making recommendations, these companies' platforms often take advantage of user accounts, to which all actions (like movie viewing, product or news surfing, purchases, and ratings) are registered; these actions are then used as input to recommendation systems [Bobadilla et al. 2013], which attempt to identify other items those users could also like.

For convenience, several people (e.g., family members or friends) may share a single account in those platforms. For example, various members of a family might share a single Amazon.com account, as it is simpler to manage all purchases in a same (already saved) bank card, and deliver those purchases to the family's address. As a result, that account history will contain actions from a group of individuals, instead of a single person.

The recommendation systems that consider users' history of activities in the platform might generate irrelevant suggestions when those are computed based on an unfiltered account history. Consider for example a shared account in a video streaming platform that one person uses to see sports-related videos and another one uses for watching comedy. In this case, resulting recommendations will be a mix of sports and comedy videos, or some other item that might be irrelevant to either person. More importantly, the recommendation may not capture the actual interests of the person currently in, and could be regarded as irrelevant; such situation could be particularly problematic for reinforcement learning based recommendation engines.

2. Motivation

The exponential growth of contents/items available in companies' platforms has made recommendation system a critical component to help users find items they might find interesting. Pathak et al. [Pathak et al. 2010] have also shown that accurate item recommendations not only improve sales revenue, but also have a higher positive effect on sales than consumer feedback. In this context, improving the quality of recommendation has become imperative to online companies. The environment in which recommendation engines operate however is complex in nature, because of (i) the sheer amount of data, users, and items to process in order to make recommendations, (ii) strict requirements for high-quality recommendations generated near real-time, (iii) customers with very few/large preference/purchase data available, among others [Linden et al. 2003].

These aspects have motivated companies to invest significantly in research towards improved recommendation accuracy. Some companies even promote/sponsor competitions in this field, like Netflix Prize [Bell and Koren 2007] and the recurring RecSys challenge[1]. They frequently assume however that actions associated with user accounts reflected individual interests, which is often not the case. In a talk, Rastogi [Rastogi 2015] mentioned that handling multiple persona behind single customer accounts is one of the research challenges pursued by Amazon.com. Although they have made progress on predicting customers' preferred product sizes in such cases [Sembium et al. 2018], unveiling those persona's profiles (comprising media, shopping, and/or news preferences and interests) remains an open problem.

[1]RecSys challenge Website: `http://www.recsyschallenge.com/2019/`.

3. Related Work

The literature on item recommendation for users behind shared accounts is incipient. Sembium *et al.* [Sembium et al. 2018] approached this problem inside Amazon.com in the context of product size recommendation for items like apparel and shoes. In their case, learning a single true size per shared account could lead to inaccurate true size estimates, therefore thwarting the authors' main goal of reducing incorrect product size purchases by customers. Their approach for handling multiple persona is introducing an array of latent variables in their model to capture multiple persona. In addition to being specific to the authors' approach, this solution is not able to deal with items having a diverse nature.

Another contribution is the work of Jiang *et al.* [Jiang et al. 2018], whose goal is identifying shared accounts in the multimedia streaming domain. Using a framework based on feature learning, unsupervised learning, and normalized random walks, the authors are able to identify a set of users behind a shared account given its music streaming sessions. In addition, given a new streaming session from an account, their framework is able to match it to an identified persona. The authors' solution is heavily dependent on feature extraction and random walk for homogeneous items (like music items), which limits its applicability in an e-commerce scenario.

Zhang et al. [Zhang et al. 2012] also focused on identifying shared accounts in the context of movie platforms. The authors' goal is (i) identifying if a given account is shared, and (ii) cluster the actions of users sharing it. They developed a model for shared accounts based on unions of linear subspaces, and applied subspace clustering to carry out the identification task. Their approach is limited to movie rating however. Finally, White *et al.* [White et al. 2014] explored custom web search on shared devices. They present methods for identifying shared devices, estimating the number of users on each device, and assigning new queries to individual user profiles. The estimated number is used to guide a *k-means* cluster in the segregation of the device search log into k person records. A new query is assigned to a cluster by applying a resource-based similarity score such as topic, time, and query size. Again, this work is limited as it cannot be applied to item recommendation in online platforms.

The proposal solution, unlike Jiang *et al.* [Jiang et al. 2018] we do not assume the nature of items visited with the potentially shared account (which can be music, shoes, apparel, and so on). Recommendation engines will compute item suggestions based on the profile better matching the set of items being visited by the person currently in (instead of the entire account history).

4. Proposed Methodology

Our approach for identifying users behind a shared account consists in building users' profiles out of users' online sessions originated from that account. Our research is built on the hypothesis that online sessions from a same person will likely be similar (e.g. in terms of visited set of items or item categories) and therefore may be grouped together into clusters. We discuss the validity of that hypothesis later in Section 5.

There are three challenges that need to be addressed for materializing our approach: (*i*) identify users' online sessions; (*ii*) represent those sessions; and (*iii*) cluster them into users' profiles. For the first one, we consider a mechanism able to capture —

as a single user session in the platform — browsing actions made by the same person in a given period. The challenge is deciding when some session ends and another one starts. One trivial case is a long period of inactivity. A more complex case, however, is when a person (who was browsing for new age music) gives seat to another one (who now listens to heavy metal). In our research, we consider item-item similarity metrics (like correlation indexes and cosine similarities between item vectors [Sarwar et al. 2001]), in addition to idleness, to delimit browsing sessions. We do so by computing an average *distance* between visited items; once it exceeds a threshold, it indicates that someone else is now browsing (a new session).

The second challenge is addressed using classical information retrieval approaches for document representation [Manning et al. 2008]. We currently represent users' sessions using a vector of terms (and their frequencies) that describe visited items. In this context, our research relates to other investigations [Kannan et al. 2011] that also use term-weighting as a resource to describe items in recommendation systems.

The third challenge requires a clustering technique that does not require an estimate for numbers of clusters that should be formed, as it is not possible to determine beforehand how many users share an account. In our work, we adopt Affinity Propagation (AP) [Frey and Dueck 2007]. In addition to not requiring a pre-determined number of clusters, AP makes no assumption of how online sessions are represented. In this context, online sessions are regarded as data points; they are grouped together around *exemplars* (sessions that are more representative of a cluster) by means of a function that quantifies session similarity. We adopted cosine similarity in our first iteration at this problem.

Another aspect to be addressed is related to sessions already registered in the account history vs. ongoing online sessions. We approach this aspect as described next. In a first moment, the account history is parsed into online sessions, which are then grouped together into clusters, with each cluster representing a user profile. Then, for each new session, they are checked against exemplars of already existing clusters. In case a high similarity is found to an exemplar, the session is aggregated to the cluster of that exemplar; otherwise, a new cluster is formed, using that session as exemplar.

5. Experiments and Preliminary Results

We carried out a preliminary evaluation using data sets from Instacart's market shopping list[2] and Last.fm music listening sessions[3]. We show a summary of the data sets in Table 1. There is no information in the data sets about account sharing. For this reason, to assess the validity of our hypothesis, using the same approach of [Jiang et al. 2018], we created synthetic "shared accounts", by randomly grouping users (25% of shared accounts) in groups of two, three or four, following a strategy used in previous investigations [Zhang et al. 2012, Verstrepen and Goethals 2015]. For the sake of data cleansing, we excluded accounts with fewer than ten entries. In the case of Last.fm, after applying that filter, we obtained 920 user accounts, which were then grouped into shared accounts as shown in Table 2.

In our evaluation, we want to answer the following question: How effectively can we uncover individual user profiles behind those shared accounts? There are two

[2] https://www.kaggle.com/c/instacart-market-basket-analysis

[3] http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html

	Period	Session/Visited items	Number of users
Instacart	2017	3,346,083	206,209
Last.fm	(Published in May, 2010)	907,887	992

Table 1. Summary of the data sets used in our evaluation.

	1	2	3	4
Instacart	4,670	9,340	14,010	18,680
Last.fm	92	184	276	368

Table 2. Number of shared accounts by quantity of users (1, 2, 3, and 4) sharing them

aspects that must be observed. First, our solution should identify as a shared account those we created after merging two or more individual accounts into them. In this case, the resulting clusters must equal the individual accounts merged. To assess error in the clustering process, i.e., comparing original users' accounts and clusters formed from the sessions contained in synthetic ones, we used Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics, as shown in Table 3. Observe that measured error increases marginally with the number of users within the shared account, suggesting that one could in fact reconstruct the original accounts using clustering. These results however also indicate that additional measures are required to further improve clustering.

	1	2	3	4
MAE	3.771	4.357	5.234	6.746
RMSE	4.192	5.043	5.767	7.261

Table 3. Values for MAE and RMSE by quantity of users (1, 2, 3, and 4), for Instacart

For the second aspect of our evaluation question, note that some user accounts from the original data sets might, in fact, be shared ones; our solution should reveal them as well, aggregating the activities of those users into separate clusters. In this case, since we do not have information about account sharing from the original data set, we evaluate the quality of resulting clusters as an indicator of the effectiveness of our solution. To this end, we used *Adjusted Rand Index* (ARI), *Adjusted Mutual Information* (AMI) and *Fowlkes-Mallows Index* (FMI), which combined may indicate clustering performance in presence of a large number of clusters. The results are shown in Tables 4 and 5.

	ARI		AMI		FMI	
User by account	AP	w/o AP	AP	w/o AP	AP	w/o AP
1	0.0	**1.0**	0.0	**1.0**	0.5131	**1.0**
2	**0.3623**	0.0	**0.3762**	0.0	0.5997	**0.7307**
3	**0.4388**	0.0	**0.4883**	0.0	**0.6125**	0.6091
4	**0.4586**	0.0	**0.5323**	0.0	**0.6053**	0.535

Table 4. Results achieved with the Instacart data set.

The Adjusted Rand Index (ARI) provides a measure of how similar the units (in our case, user sessions) within a cluster are. In this case, higher values for ARI may suggest a better performance of our solution in identifying user profiles behind potentially shared accounts. Observe in Table 4 that the clustering quality improves with the number of users per account, suggesting that clustering may be a valuable tool in aggregating

users' actions based on session similarity. In the case of Last.fm results shown in Table 5, the results indicate poor clustering performance, close to randomness, which suggests that further investigation is required.

Next we compare formed clusters using AMI; values closer to zero indicate label assignments that are largely independent, measuring the similarity between data points in clusters representing random clusters; conversely, values closer to 1 indicate clusters that are more similar. Again, observe in Table 4 that results obtained with Instacart indicate a positive trend in clustering as the number of users per shared account increases. Finally, FMI indicates similarity between groups, with perfect labeling (meaning higher quality aggregation of users' profiles in our context) is scored 1; conversely, independent labeling has score zero. The results obtained provide evidence that a more positive clustering was obtained for the Instacart dataset.

The best results database was Instacart showing that the repetition of items between sessions is important to obtain good results in the cosine similarity metric. The Last.fm database was the one that obtained the worst results, and this is due to having very long sessions. Even though there are repeats of items between them, compared to the total of items are few, which makes the cosine similarity a very low value. Given this analysis, the next step that will be performed is test other metrics that will perform other results and compare with the cosine metric.

	ARI		AMI		FMI	
User by account	AP	w/o AP	AP	w/o AP	AP	w/o AP
1	0.0	**1.0**	0.0	**1.0**	0.178	**1.0**
2	**0.0153**	0.0	**0.0826**	0.0	0.1298	**0.8049**
3	**0.0149**	0.0	**0.1166**	0.0	0.1166	**0.705**
4	**0.0146**	0.0	**0.1391**	0.0	0.1044	**0.6354**

Table 5. Results obtained with the Last.fm data set.

6. Final Considerations and Next Steps

The preliminary results obtained have suggested that clustering users' sessions is a promising direction to uncover multiple profiles behind a shared account. More specifically, the use of affinity propagation enabled us to approximate the number of users in a potentially shared account. We will follow up in our work with experiments using other data sets, considering other similarity metrics too, like Silhouette Coefficient and, for a comparative purpose another clustering method.

There are a number of aspects we are currently addressing in our research. For example, we are working on an alternate session representation scheme to include features/attributes like search strings used during navigation, browsing pattern/graph, number of items visited per unit of time, etc. In this case, a data structure suitable for near real-time, effective/efficient persistence/querying of sessions based on such attributes will become paramount. Depending on how sessions will be represented, text clustering [Manning et al. 2008], graph similarity scoring and matching [Zager and Verghese 2008] or other machine learning techniques could be used.

References

Bell, R. M. and Koren, Y. (2007). Lessons from the netflix prize challenge. *SiGKDD Explorations*, 9(2):75–79.

Bobadilla, J., Ortega, F., Hernando, A., and GutiéRrez, A. (2013). Recommender systems survey. *Know.-Based Syst.*, 46:109–132.

Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315:2007.

Jiang, J.-Y., Li, C.-T., Chen, Y., and Wang, W. (2018). Identifying users behind shared accounts in online streaming services. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 65–74, New York, NY, USA. ACM.

Kannan, A., Givoni, I. E., Agrawal, R., and Fuxman, A. (2011). Matching unstructured product offers to structured product specifications. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 404–412, New York, NY, USA. ACM.

Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Pathak, B., Garfinkel, R., Gopal, R. D., Venkatesan, R., and Yin, F. (2010). Empirical analysis of the impact of recommender systems on sales. *Journal of Management Information Systems*, 27(2):159–188.

Rastogi, R. (2015). Machine learning @ amazon. In *2nd IKDD Conference on Data Sciences*, CODS-IKDD '15, pages 2:1–2:1, New York, NY, USA. ACM.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA. ACM.

Sembium, V., Rastogi, R., Tekumalla, L., and Saroop, A. (2018). Bayesian models for product size recommendations. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 679–687.

Smith, B. and Linden, G. (2017). Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, 21(3):12–18.

Verstrepen, K. and Goethals, B. (2015). Top-n recommendation for shared accounts. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 59–66, New York, NY, USA. ACM.

White, R. W., Hassan, A., Singla, A., and Horvitz, E. (2014). From devices to people: Attribution of search activity in multi-user settings. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 431–442, New York, NY, USA. ACM.

Zager, L. A. and Verghese, G. C. (2008). Graph similarity scoring and matching. *Applied Mathematics Letters*, 21(1):86 – 94.

Zhang, A., Fawaz, N., Ioannidis, S., and Montanari, A. (2012). Guess who rated this movie: Identifying users through subspace clustering. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, pages 944–953, Arlington, Virginia, United States. AUAI Press.

Infraestrutura para Integração Semântica e Construção de *Mashup* de Dados

Matheus Mayron Lima da Cruz[1], Vânia Maria Ponte Vidal[1]

[1]Programa de Mestrado e Doutorado em Ciência da Computação (MDCC)

Universidade Federal do Ceara (UFC)

Campus do Pici – Bloco 910 – 60.455 – 760- Fortaleza– CE – Brasil

matheusmayron@alu.ufc.br, vania.maria@dc.ufc.br

Nível: Mestrado
Ingresso: Fevereiro 2018
Previsão de Término: Fevereiro 2020
Etapas já concluídas: Revisão Bibliográfica Preliminar, Definição do Problema, Defesa de Qualificação

Abstract. *In recent years, with the increase of public data available, the number of applications that use the data from these different sources has increased in order to perform an analysis of these data. However, before any analysis is conducted, it is necessary to address possible heterogeneities between the data. One of the ways to solve this problem is through the process of semantic integration of databases, creating what we call semantic view. In this way, this work presents an infrastructure to facilitate the process of creation and publication of a semantic view, besides allowing the partial or complete materialization of a semantic view.*

Keywords: *Semantic Data Integration, Data Mashup, Semantic Web*

Resumo. *Nos últimos anos, com o aumento de dados públicos disponíveis, tem crescido a quantidade de aplicações que utilizam os dados dessas diferentes fontes a fim de realizar uma análise sobre esses dados. Contudo, antes que qualquer análise seja conduzida, é necessário resolver as possíveis heterogeneidades existentes entre os dados. Uma das formas de se resolver este problema é através do processo de integração semântica das bases de dados, criando o que chamamos visão semântica. Desta forma, este trabalho apresenta uma infraestrutura para facilitar o processo de criação e publicação de uma visão semântica, além de permitir a materialização parcial ou completa de uma visão semântica.*

Palavras-chave: *Integração Semântica de Dados, Mashup de Dados, Web Semântica*

1. Introdução

A exploração e análise de dados se tornaram atividades fundamentais na descoberta de conhecimentos que podem ser utilizados no processo de tomada de decisão. Dentre os fatores que podem afetar a eficácia dessas tarefas, podemos destacar a preparação adequada dos dados e a capacidade de interpretá-los (Rahm and Do, 2000). Esses fatores são ainda mais relevantes em um contexto onde as informações estão distribuídas em fontes distintas e heterogêneas e precisam ser integradas através do processo de integração semântica.

O processo de integração semântica (Cruz and Xiao, 2005) faz uso de uma representação conceitual dos dados e de seus relacionamentos para eliminar possíveis heterogeneidades existentes entre as fontes de dados, obtendo como resultado o que definimos como visão semântica. Essa representação conceitual tem sido feita, de maneira extensiva, por meio de ontologias devido a sua capacidade de prover um conhecimento explícito e formal sobre uma conceitualização compartilhada (Guarino, Oberle and Staab, 2009). Além disso, o uso de ontologias tem sido impulsionado pela sua utilização por tecnologias da Web Semântica desenvolvidas para abordar problemas relacionados a heterogeneidade de dados em um ambiente como a Web.

A construção de uma visão semântica envolve 3 desafios principais: (1) seleção das fontes de dados relevantes para serem incluídas na aplicação; (2) definição sobre como extrair e traduzir dados provindos de fontes distintas e, possivelmente heterogêneas para um vocabulário comum; (3) definição das regras necessárias para identificar links entre recursos presentes nas diferentes fontes de dados. Mesmo que já existam pesquisas que busquem automatizar procedimentos relacionados a cada um desses desafios, a participação humana em cada uma dessas tarefas ainda é fundamental. Portanto, é necessário que existam formas de auxiliar a criação e publicação de uma visão semântica, permitindo que, uma vez especificada, uma mesma visão semântica possa ser reutilizada em mais de uma aplicação.

Após construída, a visão semântica pode ser utilizada para fornecer uma visão integrada sobre as fontes de dados por meio de dois principais enfoques: virtual e materializado. No enfoque virtual, os dados são transformados e integrados apenas no momento da consulta, permitindo que os dados sejam consumidos pelo processo integração apenas quando demandados pelo usuário, possibilitando o acesso a dados atualizados, além de não ocupar um espaço em disco desnecessário. Diferentemente, no enfoque materializado o processo de integração dos dados ocorre antes que qualquer consulta seja demandada. Este último enfoque é indicado principalmente em cenários onde se deseja realizar uma análise sobre os dados e a visão integrada precisa ser consultada de forma extensiva por consultas complexas.

Nesse contexto, este trabalho está focado no desenvolvimento de uma infraestrutura para construção de visões semânticas e de uma ferramenta para realizar a materialização parcial ou completa dos dados integrados por uma visão semântica através de um processo semiautomático. De maneira mais abrangente, essas contribuições fazem parte do desenvolvimento de um portal semântico (da Cruz *et al.*, 2019) que atua como uma ponte entre os usuários e um conjunto de aplicações e serviços que utilizam a visão semântica.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 introduz os trabalhos relacionados; a Seção 3 descreve as contribuições propostas por esse

trabalho e a seção 4 discute as conclusões parciais deste trabalho e os próximos passos a serem seguidos por esta pesquisa.

2. Trabalhos relacionados

(Calvanese *et al.*, 2016) propõe uma arquitetura OBDA para expor uma base de dados relacional como um grafo RDF. Este grafo virtual pode ser consultado por meio de consultas SPARQL através da transformação dessas consultas SPARQL em SQL sobre as bases de dados. Todo esse processo de transformação é transparente para o usuário. (Phuoc *et al.*, 2009) propõem DERI Pipes, uma ferramenta que utiliza o conceito de *semantic pipes* para fornecer uma forma de quebrar, em nível operacional, a tarefa de integração e processamento dos dados em passos menores que podem ser combinados livremente. De forma similar, (Knap *et al.*, 2014) apresentam UnifiedViews, um framework *Extract-Transform-Load* (ETL) que busca estabelecer uma forma simples de definir um processo de . Os autores permitem que processo de ETL seja visto como uma pipeline de processamento de dados formado por uma ou mais unidades de processamento (DPU) e fluxo de dados entre as DPUs.

(Schultz *et al.*, 2011) apresentam LDIF, um framework para realizar a integração de dados de bases *linked data*. O LDIF possui as etapas de tradução dos dados, resolução de identidade, fusão dos dados, além de considerarem alguns aspectos relacionados a proveniência. Contudo, não apresenta uma forma de publicar a especificação feita utilizando as tecnologias web, o que dificulta sua manipulação e a reutilização das integrações feitas por outras aplicações. Além disso, não estabelece uma formalização, uma visão holística, sobre a integração realizada entre as bases de dados. (Cavalcante et al., 2017) propõem uma abordagem que inspirou o planejado esta dissertação. Entretanto, a construção de um ambiente semântico não faz parte de (Cavalcante et al., 2017). Além disso, o método de construção de *mashups* baseado em uma visão semântica não possui algumas funcionalidades que consideramos importante, e.g., não é possível selecionar apenas um subconjunto de propriedade de determinada classe.

De maneira geral, nenhum dos trabalhos relacionados apresenta um ambiente capaz de fornecer uma visão conceitual do processo de integração semântica realizado ou uma forma de facilitar a publicação e reutilização da visão semântica estabelecida ou de seus componentes. Mesmo considerando que a utilização dos conceitos *semantic pipes* e "unidades de processamento" como importantes abstrações a nível operacional, elas são insuficientes para estabelecer uma visão holística sobre o processo de integração semântica realizado. Ademais, com exceção de (Cavalcante et al., 2017), nenhuma das abordagens se preocupa em facilitar a materialização de apenas parte da visão semântica estabelecida.

3. Contribuição Proposta

A Figura 1(a) mostra a arquitetura utilizada pelo portal semântico *SemanticSUS* (da Cruz *et al.*, 2019). A partir de um enfoque que combina ontologias e dados interligados, o SemanticSUS visa oferecer uma visão semântica sobre as fontes de dados, independente do enfoque que será utilizado na camada de integração de dados. O presente trabalho busca desenvolver dois pontos relacionados a esta arquitetura:

(1) **Ambiente de integração semântica:** Como uma das propostas desta dissertação está o desenvolvimento de ambiente para facilitar o processo de criação e

publicação de uma visão semântica. Este ambiente compõe a camada de integração semântica e visa permitir que o processo de criação ocorra de forma *pay-as-you-go*(Madhavan *et al.*, 2007). Apesar de já existirem ferramentas que auxiliam na definição de algumas das especificações necessárias para criação de uma visão semântica, como o editor de mapeamentos Map-On (Sicilia, Nemirovski and Nolle, 2017), nenhuma delas organiza essas especificações a fim de estabelecer uma visão semântica. Este ambiente compreende também o desenvolvimento de um modelo RDF capaz de descrever a visão semântica estabelecida.

(2) **A ferramenta *MashupBuilder*:** A principal contribuição proposta por este trabalho é a ferramenta *MashupBuilder*. Essa ferramenta tem objetivo permitir que a visão semântica especificada seja utilizada na construção e materialização de uma *mashup* de dados.

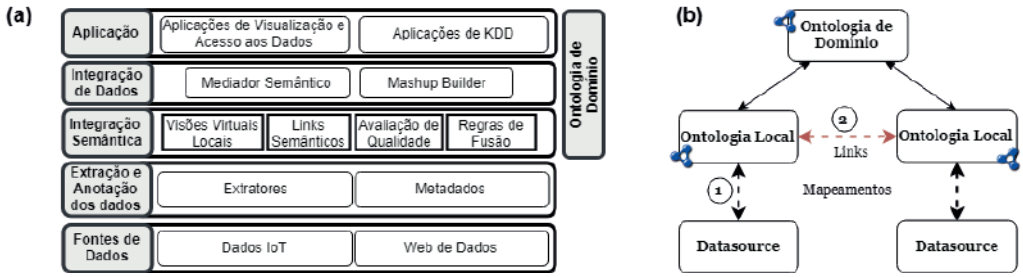


Figura 1: (a) Arquitetura do Portal Semântico e (b) Framework de Integração Semântica

3.1 Processo de Criação da Visão Semântica

Uma visão semântica pode ser definida como uma n-tupla $\lambda = (S, O_D, V, L)$, onde:

- S é um conjunto de fontes de dados $S_1, \ldots, S_n$;
- O_D é a ontologia de Domínio. Esta ontologia é responsável por estabelecer um vocabulário a ser compartilhado entre as fontes de dados semanticamente integradas;
- V é um conjunto de especificações de visões virtuais locais $V_1, \ldots, V_n$. Cada visão virtual local é composta por uma ontologia local O_{Li}, o esquema S_i de uma das bases em S um conjunto de mapeamentos M_i que relacionam os termos do esquema da base S_i aos termos da ontologia local O_{Li};
- L é um conjunto de regras de *linkage* definidas entre as classes semanticamente semelhantes das ontologias locais. Essas regras buscam estabelecer formas de reconhecer que dois recursos referenciam o mesmo objeto no mundo real, apesar de possuírem representações distintas.

A Figura 1(b) mostra o framework utilizado na construção da visão semântica em (da Cruz *et al.*, 2019). A inclusão de uma nova fonte de dados S_k em uma visão semântica existente segue os seguintes passos: (1) Especificação da visão virtual local $V_k = (O_k, S_k, M_k)$; (2) Especificação das Regras de *Linkage* que serão utilizadas para gerar links *owl:sameAs* entre as instâncias de V_k e as instâncias das demais visões virtuais locais.

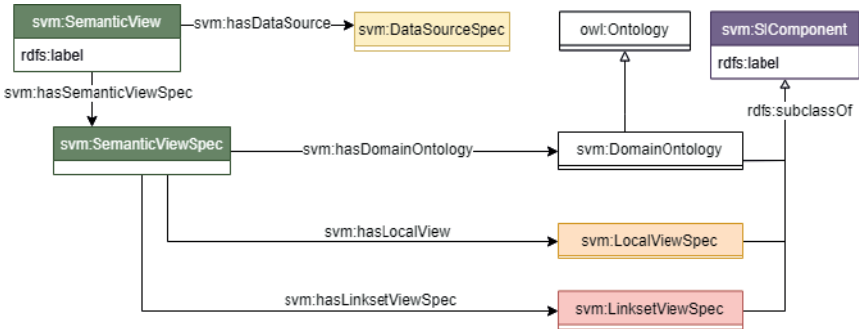


Figura 2: Recorte do Vocabulário para Representação da Visão Semântica

Baseado nesta definição de visão semântica, estamos desenvolvendo o modelo *Semantic View Modeling(*SVM*)*. Este modelo tem como objetivo ser uma representação conceitual para especificação de visões semânticas utilizando o formato RDF. Essa representação em alto nível possibilita, por exemplo, a adição de regras para checar a validade de uma visão semântica construída ou a inferência de relacionamentos entre alguns de seus elementos, além de facilitar a sua publicação e compartilhamento por adotar o formato RDF. A Figura 2 mostra uma visão geral do SVM.

O SVM também tem como objetivo permitir que a visão e seus componentes recebam comentários e descrições, com a finalidade de facilitar o entendimento de outras pessoas sobre a visão construída e sobre os elementos que a compõem. Ademais, com a finalidade de auxiliar a utilização do SVM, ainda no contexto dessa pesquisa, planejamos desenvolver um ambiente para permitir a construção e gerenciamento de uma visão semântica e de seus recursos.

3.2 Criação de *Mashups* Especializados

A criação e materialização de um *mashup* de dados é uma das formas de se fornecer acesso aos dados integrados por uma visão semântica. Um *mashup* de dados se trata de uma aplicação que oferece uma nova funcionalidade através da combinação, agregação e transformação de dados provindos de diferentes fontes. A criação de um *mashup* é essencial quando a qualidade dos dados e a velocidade de execução de consultas complexas são requisitos das tarefas que iram utilizar a visão integrada.

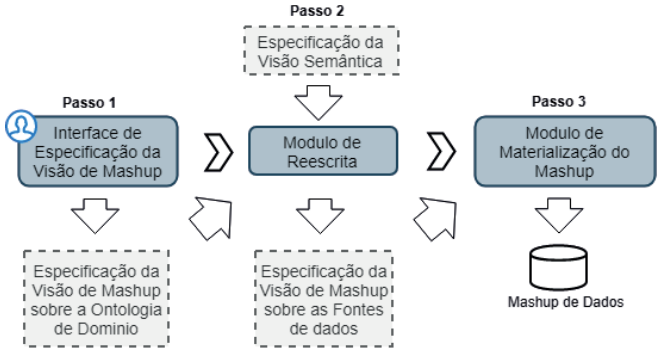


Figura 2: Processo Seguido pelo *MashupBuilder*

Neste contexto, o *MashupBuilder* é uma ferramenta que tem como objetivo semiautomatizar o processo de construção de *mashups* de dados especializados a partir de uma visão semântica especificada *a priori*. A Figura 2 mostra os passos empregados pelo *MashupBuilder*.

No primeiro passo, a especificação da visão de *mashup*, o usuário é responsável por especificar o que deve estar contido no *mashup* de dados requisitado. Essa especificação é feita baseada na ontologia de domínio. Ademais, também é necessário realizar a definição das regras de fusão que serão aplicadas sobre os dados que representam o mesmo objeto no mundo real e foram identificados por alguma das regras em L.

O resultado deste passo é chamado de visão de *mashup* sobre a ontologia de domínio e representada como uma n-tupla $VMO = (O, O_{interest}, \theta, F)$, onde: O é a ontologia de domínio na qual a visão está baseada; $O_{interest}$ é um conjunto de termos de interesse especificados pelo usuário; θ é um conjunto de filtros sobre os termos de interesse definidos em $O_{interest}$; Por fim, F é um de regras de fusão definidas pelo usuário.

No passo seguinte, é feita a reescrita da especificação da visão semântica. A partir de uma visão semântica $\lambda = (S, O_D, V, L)$ especificada *a priori*, o modulo de reescrita irá "recortar" esta visão λ, selecionando apenas o necessário de para construção do *mashup* especificado por VMO. Além disso, os filtros especificados pelo usuário serão embutidos nos mapeamentos de cada umas das visões locais em V. O resultado deste passo recebe o nome de visão de *mashup* sobre as fontes de dados e pode ser formalmente definido como uma n-tupla $VMD = (S, O_{interest}, V_{frag}, L_{frag}, F)$.

Por fim, no último passo, é feita a materialização de VMD. Esta materialização segue os seguintes passos: (1) Materialização das visões locais V_{frag}; (2) Materialização dos links descobertos pelas regras de L_{frag}; (3) Consolidação do *mashup* de dados através da aplicação das regras de fusão de dados F.

4. Conclusões Parciais e Trabalhos Futuros

Este trabalho propõe uma infraestrutura para facilitar a criação e publicação de uma visão semântica. Ademais, a ferramenta MashupBuilder também é proposta com a finalidade de, a partir de uma visão semântica especificada *a priori*, viabilizar a construção e materialização de *mashups* de dados especializados. Essas contribuições irão integrar o SemanticSUS, um portal semântico criado com a intenção de oferecer uma visão semântica sobre as fontes de dados do SUS. Atualmente, o *SemanticSUS* integra três bases de dados do SUS que disponibilizada pela plataforma GISSA.

O cronograma para finalizar essa dissertação inclui a conclusão da modelagem do *SVM* e a implementação do ambiente de criação da visão semântica. Além disso, é necessário formalizar o processo de reescrita da visão semântica a partir da especificação da visão de *mashup* e implementar a ferramenta *MashupBuilder*.

Referências

Calvanese, D. *et al.* (2016) 'Ontop: Answering SPARQL queries over relational databases', *Semantic Web*. Edited by Ó. Corcho, 8(3), pp. 471–487. doi: 10.3233/SW-160217.

Cavalcante, G. M. L., Vidal, V. M. P. and Oliveira, A. M. B. de (2017) *MAURA: Um framework baseado em Mediador Semântico para Construção Eficiente de Linked Data Mashups*. Instituto Federal de Educação, Ciência e Tecnologia do Ceará.

Cruz, I. F. and Xiao, H. (2005) 'The Role of Ontologies in Data Integration', *Science*, 13(4), pp. 1–18. doi: 10.1.1.60.4933.

da Cruz, M. M. *et al.* (2019) 'SemanticSUS: Um Portal Semântico baseado em Ontologias e Dados

Interligados para Acesso, Integração e Visualização de Dados do SUS', in *Anais Estendidos do XIX Simpósio Brasileiro de Computação Aplicada à Saúde*. Porto Alegre, RS, Brasil: SBC, pp. 13–18. Available at: https://sol.sbc.org.br/index.php/sbcas_estendido/article/view/6277.

Guarino, N., Oberle, D. and Staab, S. (2009) 'What Is an Ontology?', in *Handbook on Ontologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–17. doi: 10.1007/978-3-540-92673-3_0.

Knap, T. *et al.* (2014) 'UnifiedViews: An ETL framework for sustainable RDF data processing', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8798, pp. 379–383. doi: 10.1007/978-3-319-11955-7_52.

Madhavan, J. *et al.* (2007) 'Web-scale Data Integration: You can only afford to Pay As You Go', pp. 342–350.

Phuoc, D. Le *et al.* (2009) 'DERI Pipes : visual tool for wiring Web data sources', *Proceedings of the 18th international conference on World wide web - WWW '09*, (March 2014), p. 581. doi: 10.1145/1526709.1526788.

Rahm, E. and Do, H.-H. (2000) 'Data Cleaning: Problems and Current Approaches | Dokumentenserver', *IEEE Bulletin of the Technical Committee on Data Engineering*, 23(4), p. 11. Available at: http://lips.informatik.uni-leipzig.de/pub/2000-45.

Schultz, A. *et al.* (2011) 'LDIF -Linked Data Integration Framework', *CEUR Workshop Proceedings*, 782, pp. 1–6. doi: urn:nbn:de:0074-782-7.

Sicilia, Á., Nemirovski, G. and Nolle, A. (2017) 'Map-On: A web-based editor for visual ontology mapping', *Semantic Web*. Edited by K. Janowicz, 8(6), pp. 969–980. doi: 10.3233/SW-160246.

Mineração de Sequências Restritas no Espaço e no Tempo

Aluno: Antonio Jose de Castro Filho[1]
Orientadores: Eduardo Ogasawara[1]**, Rafaelli Coutinho**[1]

[1]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ

antonio.castro@eic.cefet-rj.br, eogasawara@ieee.org, rafaelli.coutinho@cefet-rj.br

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ
Programa de Pós-Graduação em Ciência da Computação - PPCIC
Ingresso no Programa de Pós-Graduação: março de 2018
Previsão de Defesa do Mestrado: março de 2020

Etapas Concluídas: Qualificação.

Etapas a concluir: Avaliação Experimental/Ajustes na Metodologia, Defesa.

Abstract. *Given a dataset containing events with time and location tags, in the middle of such a dataset it is possible to give existence of sequential patterns, that is, it is observed that the event B always occurs after another event A, at nearby locations, which may indicates a relation of consequence, cause and effect. The discovery of such patterns can be useful in a wide variety of domains for predicting future events, in order to support decision making and plan future actions. Some patterns, however, are not common all the time and / or in every space. The Mining of Restricted Sequences in Space and Time search for sequences that may be under-supported if considered the entire dataset, but may be relevant and strongly supported in some spatio-temporal partition.*

Resumo. *Dado um conjunto de dados contendo eventos com marcações de tempo e de localização, em meio a tal conjunto é possível se dar a existência de padrões sequenciais, ou seja, observa-se que o evento B ocorre sempre após um outro evento A, em localizações próximas, o que pode indicar uma relação de consequência, causa e efeito. A descoberta de tais padrões pode ser útil em uma grande diversidade de domínios para a previsão de eventos futuros, de forma apoiar a tomada de decisões e planejar futuras ações. Alguns padrões, contudo, não são comuns todo o tempo e/ou em todo espaço. A Mineração de Sequências Restritas no Espaço e no Tempo busca por sequências que talvez apresentem baixo suporte se considerado todo o conjunto de dados, mas podem ser relevantes e com grande suporte em alguma partição espaço-temporal.*

1. Introdução

Novas tecnologias, como telefones celulares e sensores digitais, tornam cada vez mais comum a geração de grandes bases de dados com marcações de espaço e tempo. Registros de corridas de táxi, rotas utilizadas por usuários enquanto dirigem seus carros, são alguns desses novos conjuntos de dados. Existe uma grande quantidade de dados disponíveis acerca de diversas áreas de conhecimento. Tais conjuntos de dados possuem em si padrões de ocorrência de eventos e registro do comportamento de usuários em um dado momento e em uma dada posição. A análise destes dados pode fornecer conhecimentos úteis [Huang et al., 2008].

Ser capaz de obter conhecimento de padrões existentes nessas bases de dados é um diferencial importante. A informação da ocorrência de um evento A poder prever ou quantificar a probabilidade de um evento B é de grande utilidade para tomada de decisões em diversos domínios. Por exemplo, para obter conhecimento dos padrões de compras de uma rede de lojas e dos dados de viagens de uma companhia aérea ou de um aplicativo de corridas de táxi. Essas informações abrem espaço para busca de padrões espaço-temporais [Alatrista-Salas et al., 2016; Li and Fu, 2014; Yan Huang et al., 2008; Klemettinen et al., 1994].

Para alguns desses domínios é difícil, ou mesmo impossível, a avaliação em busca de padrões de forma não automatizada dado o tamanho de suas bases de dados. Por exemplo, uma rede de supermercados apresentaria grande dificuldade de analisar todos os padrões de compras de suas filiais de forma manual. O uso de mineração de dados, dentro de um processo multidisciplinar, como forma de descoberta de estruturas de interesse em grandes conjuntos de dados torna a análise desses dados possível [Hand, 2007; Fayyad et al., 1996].

Para atingir o intuito da descoberta de padrões, algoritmos de Mineração de Dados têm sido aplicados em uma grande diversidade de problemas. Esses algoritmos começaram com a busca de regras de associação e evoluíram para o tratamento de mineração de padrões sequenciais [Agrawal et al., 1993; Agrawal and Srikant, 1995]. Com novas bases de dados comportando marcações de espaço e tempo disponíveis, a busca por padrões passa a análise dos relacionamentos de espaço e tempo dos dados coletados em busca de padrões espaço-temporais [Han et al., 2007].

Nem sempre a frequência de ocorrência de alguns padrões é grande se considerada em todo o conjunto de dados. Surge, então, a ideia de extrair não só os eventos que sejam frequentes, mas também especificar o período de tempo e uma região do espaço ao longo do qual, um dado conjunto de eventos é frequente [Saleh and Masseglia, 2008]. Este trabalho tem por objetivo principal propor uma solução para o processo de Mineração de Dados Espaço-Temporais, que busca encontrar padrões restritos em uma região no espaço e no tempo. Desta forma, buscam-se não só as sequências que são padrões, mas também o intervalo de tempo e de espaço onde tais sequências são frequentes.

As principais contribuições deste trabalho podem ser resumidas nos seguintes pontos: (i) detalhar os conceitos e os passos da busca por padrões de sequências em séries espaço-temporais; (ii) propor um algoritmo que solucione o problema; e (iii) disponibilizar a implementação do algoritmo e realizar a execução de experimentos.

O restante deste trabalho é organizado como segue. A Seção 2 introduz co-

nhecimentos importantes para a compreensão deste trabalho. A Seção 3 apresenta a classificação dos trabalhos relacionados a este. A Seção 4 descreve a proposta para a solução do problema abordado. Finalmente, a Seção 5 mostra uma visão geral do andamento do trabalho.

2. Fundamentos

Algumas definições importantes e necessárias para a compreensão deste trabalho são descritas nessa seção. Seja $t = <v_1, v_2, \cdots, v_n>$ uma sequência de itens com marcação de tempo (por simplicidade sequência), onde $|t| = n$ é o número de itens em t. Um índice temporal j é um valor inteiro entre 1 e n que está relacionado ao item v_j.

Uma posição p é uma dupla (x, y), onde x e y indicam posições em um sistema de coordenadas (neste trabalho o espaço é restrito a duas dimensões). Seja $P = \{p_1, p_2, \ldots, p_m\}$ um conjunto de posições. Uma **sequência de itens com marcação de tempo e espaço (STS)** d é uma dupla (p, t) onde $p \in P$ é uma posição e t é a marcação de tempo associada. Desta forma, um conjunto de dados de STS D é um conjunto sequências de itens com marcação de tempo e espaço.

Uma sequência $s = <w_1, w_2, \cdots, w_k>$ está contida em outra sequência $z = <v_1, v_2, \cdots, v_n>$ se existirem inteiros $i_1 < i_2 < \cdots < i_k$ tais que $w_1 = v_{i_1}, w_2 = v_{i_2}, \cdots, w_k = v_{i_k}$. Diz-se que uma STS $d = (p, t)$ suporta uma sequência s se s está contida em t. O suporte de uma sequência s em D é igual ao número de STS em D no qual s está contida.

A frequência de uma sequência s em D é a fração de STS em D que suportam s: $freq(s, D) = \frac{sup(s,D)}{|D|}$. Dado um valor mínimo, definido pelo usuário $\gamma \in]0..1]$, uma sequência é dita frequente se $freq(s, D) \geq \gamma$. Seja I o conjunto de todas as possíveis sequências s contidas em D. O conjunto F de sequências frequentes em D considerando γ é denotada por $F(D, \gamma) = \{s \in I : freq(s, D) \geq \gamma\}$.

Sejam f e h duas posições, tais que, $f = (x_f, y_f)$ e $h = (x_h, y_h)$. A distância entre f e h é dada por $dist(f, h) = \sqrt{(x_h - x_f)^2 + (y_h - y_f)^2}$, ou seja, a distância euclidiana. Um grupo de posições espaciais (por simplicidade grupo) G é definido por um conjunto de duas ou mais posições onde seus elementos devem estar a uma distancia máxima σ de ao menos um outro elemento do grupo, ou seja, $G = \{p \in P : \exists q \in G, dist(p, q) \leq \sigma\}$. Define-se o conjunto de todos os potenciais grupos sobre D como PG. O conjunto de STS que pertencem a um grupo G é definido como $Tr(G) = \{d : d \subseteq D, d.p \in G\}$

A frequência de s sobre $Tr(G)$ em STS de um grupo r é denotada por $freq(s, G)$. A mesma abordagem é válida para o suporte de s sobre $Tr(G)$, o qual é denotado por $sup(s, G)$. Uma sequência espacial sr é um trio (s, G, fr), onde s é uma sequência, G é um grupo, e fr é a frequência da sequência s sobre o grupo G: $fr = freq(s, G)$.

Um intervalo de tempo (por simplicidade intervalo) $i = (i_i, i_f)$ é definido por uma marcação de tempo inicial i_i e uma marcação de tempo final i_f. O tamanho do intervalo i é dado por: $|i| = i_f - i_i + 1$. Seja s uma sequência e k o tamanho de s, então s é chamada uma sequência de tamanho k. Dado um intervalo i, uma sequência $s = <w_1, w_2, \cdots, w_k>$ é uma subsequência de outra sequência $t = <v_1, v_2, \cdots, v_n>$: $s = subseq(t, i)$ se e somente se $i_i \geq 1 \wedge i_f \leq n$, $|i| = k$ and $\forall j \in [1..k], w_j = v_{i_i+j-1}$. Por definição, s também está contida em t. Define-se o conjunto de todas os possíveis

intervalos de tempo sobre D como PI.

Um bloco b é um par (G, i) onde G é um grupo ($G \in PG$) e i é um intervalo ($i \in PI$). O tamanho do bloco b é o produto do número de posições do bloco pelo tamanho do intervalo: $|b| = |b.G| \cdot |b.i|$. O conjunto de todos os possíveis blocos sobre um grupo G é definido como $PB(G)$.

3. Trabalhos Relacionados

Este trabalho foi realizado a partir do desenvolvimento de um mapa sistemático, baseado em pesquisa na base de dados *Scopus*, buscando por palavras chave relacionadas ao tema, utilizando a seguinte busca: ("*sequence mining*" *or* "*sequential pattern*") *and* ("*space-time*" *or* "*spatio-temporal*"). Os resultados foram limitados à artigos na língua inglesa, o que forneceu um total de quinhentos e quarenta e cinco documentos. A maioria dos artigos encontrados não são relacionados ao tema de mineração de dados.

Após a leitura de alguns dos artigos, foi possível classificá-los de acordo com as bases de dados que utilizam em suas pesquisas. Alguns artigos utilizam bases de dados de trajetória, outros utilizam bases de dados relacionadas a eventos, alguns outros artigos trabalham sobre bases de dados que são continuamente alimentadas, onde frequentemente surgem novos padrões ou padrões existentes são alterados.

As bases de dados de trajetórias descrevem uma coleção de eventos do mesmo objeto em movimento em diferentes marcações de espaço e tempo [Aydin and Angryk, 2016]. Buscam encontrar padrões de deslocamento de objetos em movimento [Huang et al., 2008]. Estes tipos de trabalho, embora importantes, buscam padrões diferentes dos que são procurados na abordagem adotada neste trabalho.

Em bases de dados que são alimentadas de maneira contínua, é importante observar a possibilidade de mudança de padrões. Padrões identificados inicialmente em um conjunto de dados podem ser alterados, ou deixar de existir em uma versão mais recente do conjunto de dados [Tsai and Shieh, 2009]. Este tipo de pesquisa é ortogonal a utilizada neste trabalho.

Nos trabalhos que utilizam conjuntos de dados baseados em eventos, o objetivo é encontrar sequências restritas no espaço e tempo relacionadas a eventos [Tsoukatos and Gunopulos, 2001]. É nesta categoria que se enquadra este trabalho. Apesar disto, a maioria dos trabalhos relacionados buscam apenas por sequências frequentes por todo o conjunto de dados, enquanto o presente trabalho busca por sequências frequentes apenas em intervalos de tempo e regiões no espaço considerado em duas dimensões. O trabalho de Campisano et al. [2018] é o único relacionado ao mesmo tema, contudo, considera o espaço de forma linear, enquanto o presente trabalho o generaliza considerando uma região planar.

4. Proposta

O objetivo principal deste trabalho é minerar sequências no espaço e no tempo. Desta forma, busca-se por sequências que talvez não apresentem um grande número de ocorrências por toda a base de dados, mas que são frequentes em um determinado intervalo de tempo e conjunto de posições. Para atingir tal objetivo foi desenvolvido o Algoritmo 1, chamado de STSM-3D. Ele recebe como entrada um conjunto de dados

espaço-temporal (D), um conjunto de posições (P), uma frequência temporal mínima (γ), um tamanho mínimo para o grupo (β) e uma distância máxima para os elementos do grupo (σ). O STSM-3D fornece como retorno sequências restritas no espaço e no tempo, de tamanho crescente $(2, 3, ..., n)$, que atendam as restrições definidas.

Algoritmo 1: STSM-3D

Entrada: $D, P, \gamma, \beta, \sigma$

```
1 início
2     transacoes ← traduzir(D)
3     blocos ← gerarBlocos(transacoes, P, γ, β, σ)
4     candidatos ← gerarCandidatos(blocos)
5     enquanto (candidatos ≠ ∅) faça
6         blocos, sequencias_k ←
            checarCandidatos(candidatos, transacoes)
7         candidatos ← gerarCandidatos(blocos)
8     fim
9 fim
```

Saída: $sequencias$

O procedimento *traduzir* converte a base de dados de um formato tabular para o formato de transações. No formato tabular cada coluna representa uma **STS** e as linhas representam as marcações de tempo. Ele percorre as linhas do conjunto de dados em forma tabular, verificando para cada linha (referentes às marcações de tempo) em quais colunas (as posições) cada sequência ocorre. Como resultado obtém-se o conjunto de dados no formato de transações, onde cada registro possui a sequência, a marcação de tempo, e as posições onde este ocorre.

Partindo da base de dados no formato de transações a geração de blocos (procedimento *gerarBlocos*) verifica, para cada sequência, intervalos de tempo no qual ocorre com frequência maior ou igual a γ. Para cada intervalo de tempo, utilizam-se as posições para formação de grupos, respeitando uma distância menor ou igual a σ para os componentes do grupo. O intervalo de tempo encontrado e cada um dos grupos formados geram um ou mais blocos para cada sequência. Os grupos com número de elementos menores β são desconsiderados.

A geração de candidatos (procedimento *gerarCandidatos*) segue o princípio do algoritmo Apriori, em que conjunto de candidatos de tamanho k é gerado a partir da junção de elementos frequentes em $k-1$ e, para os candidatos de tamanho k, os que possuírem em sua formação elementos que não são frequentes em $k-1$ são removidos. O procedimento faz o produto cartesiano da lista contendo sequências e blocos com ela mesma, gerando assim todas as possibilidades de candidatos. A redução da lista de candidatos é alcançada através da verificação das seguintes condições: (i) suporte no tempo; (ii) coincidência no espaço; e (iii) antimonotocidade.

O suporte no tempo mostra que para ocorrer uma sequência, é necessário que o bloco que contém a primeira parte ocorra em um período de tempo compatível com o da segunda parte. Por exemplo uma sequência AB não pode ser gerada se o intervalo de tempo de A for $[3, 5]$ e o de B for $[1, 2]$, em outras palavras, não é possível gerar

uma sequência AB se B ocorre antes de A. Usando o mesmo exemplo, observe-se que é possível gerar a sequência BA. A coincidência no espaço trata da necessidade das posições onde os blocos ocorrem terem de coincidir para que seja possível gerar candidatos. Por fim, antimonotocidade se relaciona a necessidade de que todas as subsequências têm que ser frequentes para que uma sequência o seja.

O procedimento *checarCandidatos* consiste em verificar a real ocorrência dos candidatos. Como resultado, obtém-se sequências e respectivos blocos que serão usadas para a geração de candidatos de tamanho $k + 1$ e as sequências que realmente ocorrem, com seus respectivos tempos e posições de ocorrência . O algoritmo termina quando não existirem novas sequências candidatas a serem exploradas.

A seleção dos parâmetros referentes a distância máxima dos componentes do grupo (σ) e o tamanho mínimo para o grupo (β) deve ser criteriosa e levar em conta as posições (P) das séries temporais e o que o usuário deseja alcançar. Por exemplo, um valor pequeno para σ, para a qual poucos elementos possam ser agrupados, alinhada a um grande valor de β, pode gerar a exclusão de grande número de blocos, revelando apenas padrões que ocorram bem próximos e em grande quantidade. Um valor baixo de γ também influencia bastante nos resultados, pois tende a permitir blocos com maior intervalo de tempo e com maior número de posições, diminuindo o número de grupos desconsiderados.

5. Estado Atual do Trabalho

O objetivo principal deste trabalho já foi atingido, existe um processo proposto como solução do problema de Mineração Dados Espaço-Temporais, a qual busca por padrões restritos no espaço e no tempo. Além disso, gerou-se um algoritmo capaz de realizar tal tarefa. Existe ainda uma implementação do algoritmo em linguagem R, porém em processo de otimização.

Uma versão da implementação encontra-se disponível no sítio *GitHub* em dois formatos: um *script* e um *Jupyter Notebook*, com um exemplo didático, ambos com a mesma implementação [Castro, 2019]. Em ambos é utilizado o mesmo conjunto de dados e posições como exemplo. Uma vez que a implementação em linguagem R esteja otimizada, e devidamente testada, ela será utilizada em avaliações experimentais e então disponibilizada para uso da comunidade científica como um pacote em linguagem R.

A dissertação referente a este trabalho encontra-se em processo de escrita, necessitando de maior detalhamento na formalização do processo, alguns ajustes na metodologia e a execução de experimentos para que o capítulo referente a avaliação experimental seja devidamente escrito, em especial a avaliação do desempenho do algoritmo. Por fim, pretende-se também a submissão de um artigo a um periódico.

Referências

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM.

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE.

Alatrista-Salas, H., Bringay, S., Flouvat, F., Selmaoui-Folcher, N., and Teisseire, M. (2016). Spatio-sequential patterns mining: Beyond the boundaries. *Intelligent Data Analysis*, 20(2):293–316.

Aydin, B. and Angryk, R. A. (2016). Spatiotemporal event sequence mining from evolving regions. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 4172–4177. IEEE.

Campisano, R., Borges, H., Porto, F., Perosi, F., Pacitti, E., Masseglia, F., and Ogasawara, E. (2018). Discovering tight space-time sequences. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 247–257. Springer.

Castro, A. J. (2019). Space and time constrained sequence mining - stsm-3d. `https://github.com/castroantonio/STSM-3D`.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., et al. (1996). Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88.

Han, J., Cheng, H., Xin, D., and Yan, X. (2007). Frequent pattern mining: current status and future directions. *Data mining and knowledge discovery*, 15(1):55–86.

Hand, D. J. (2007). Principles of data mining. *Drug safety*, 30(7):621–622.

Huang, Y., Zhang, L., and Zhang, P. (2008). A framework for mining sequential patterns from spatio-temporal event data sets. *IEEE Transactions on Knowledge and data engineering*, 20(4):433–448.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I. (1994). Finding interesting rules from large sets of discovered association rules. In *Proceedings of the third international conference on Information and knowledge management*, pages 401–407. ACM.

Li, K. and Fu, Y. (2014). Prediction of Human Activity by Discovering Temporal Sequence Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1644–1657.

Saleh, B. and Masseglia, F. (2008). Time aware mining of itemsets. In *2008 15th International Symposium on Temporal Representation and Reasoning*, pages 93–97. IEEE.

Tsai, C.-Y. and Shieh, Y.-C. (2009). A change detection method for sequential patterns. *Decision Support Systems*, 46(2):501–511.

Tsoukatos, I. and Gunopulos, D. (2001). Efficient mining of spatiotemporal patterns. In *International Symposium on Spatial and Temporal Databases*, pages 425–442. Springer.

Yan Huang, Liqin Zhang, and Pusheng Zhang (2008). A Framework for Mining Sequential Patterns from Spatio-Temporal Event Data Sets. *IEEE Transactions on Knowledge and Data Engineering*, 20(4):433–448.

Processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um *data warehouse* de imagens em ambientes paralelos e distribuídos

Guilherme Muzzi da Rocha[1],
Profa. Dra. Cristina Dutra de Aguiar Ciferri[1]

[1]Pós-Graduação em Ciências de Computação e Matemática Computacional
Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos, SP, Brasil

guilherme.muzzi.rocha@usp.br, cdac@icmc.usp.br

Nível: Mestrado
Ano de ingresso no programa: 2018
Exame de qualificação: Abril de 2019
Época esperada de conclusão e defesa: Abril de 2020
Etapas Concluídas: Créditos em disciplinas; Definição do Problema; Proposta de solução; Testes de desempenho preliminares
Publicações: [Rocha and Ciferri 2018, Traina et al. 2019]

Abstract. *Analytical queries in conventional data warehousing environments have a high computational cost, as they run over voluminous data warehouses and require the processing of expensive star join operations. This cost is even greater in image data warehousing environments. First, image data warehouses are more voluminous. Second, analytical queries are extended with a similarity search predicate, which requires the processing of costly operations that calculate the distance between images. In the literature, the use of parallel and distributed computing environments has become an attractive alternative to individually minimize the cost of the star join processing over conventional data and the cost of calculating the distances between images. In our master's research, we fill a gap in the literature by jointly investigating the processing of analytical queries extended with a similarity search predicate over image data warehouses using the framework Spark. In this paper, we describe the methods that we are developing to this end. We consider the context of medical images, due to the importance of the analytical decision-making over these images and their impact on society*[1].

Palavras-Chave. *Data warehouse de imagens, consultas analíticas estendidas com predicado de similaridade, Spark, imagens médicas.*

[1]Trabalho sendo desenvolvido com recursos financeiros da FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), processos 2018/10607-3 e 2018/22277-8, e do CNPq.

1. Introdução

Um ambiente de *data warehousing* engloba técnicas e ferramentas voltadas à extração, tradução, filtragem e integração de dados de provedores autônomos, heterogêneos e distribuídos (processo ETL - *extract, transform, load*). Esses dados são armazenados no *data warehouse* (DW), que é um banco de dados caracterizado por ser integrado, não-volátil, orientado a assunto e histórico. Sobre o DW incidem as consultas analíticas (processo OLAP - *on-line analytical processing*) [Kimball and Ross 2002]. Um DW convencional contém apenas dados convencionais, como dados do tipo numérico, alfanumérico e data.

Um ambiente de *data warehousing* de imagens estende o *data warehousing* convencional para também manipular imagens representadas por seus vetores de características e atributos para pesquisa por similaridade [Teixeira et al. 2015]. O processo ETL é estendido para também extrair as características intrínsecas das imagens, o DW de imagens estende o DW convencional para armazenar essas características e o processo OLAP também oferece suporte para consultas analíticas estendidas com predicado de similaridade de imagens. Como resultado, uma nova gama de consultas analíticas pode ser realizada. Por exemplo, em uma aplicação médica, pode-se determinar *"Quantas imagens de câncer pulmonar são similares a uma determinada imagem e pertencem a pacientes com idade maior do que 40 anos no estado de São Paulo nos últimos 3 anos"*.

Um DW convencional é muito volumoso. Ele é frequentemente povoado, além de conter dados históricos [Kimball and Ross 2002]. Consultas OLAP são caras porque a junção-estrela tem custo computacional alto por lidar com tabelas de fatos muito volumosas em implementações relacionais. Um DW de imagens é ainda mais volumoso porque contém dados convencionais e características intrínsecas das imagens. A frequência de povoamento também é alta. Por exemplo, em uma aplicação de uma rede de hospitais de uma determinada região, os provedores podem ser numerosos e as atividades da área médica muito frequentes [Sebaa et al. 2018]. Consultas OLAP estendidas com predicado de similaridade de imagens são muito mais caras porque envolvem onerosos cálculos de distância entre imagens [Traina et al. 2007] em adição à operação de junção-estrela.

Um *data warehousing* de imagens pode se beneficiar de ambientes computacionais paralelos e distribuídos. O uso desses ambientes tem se tornado uma alternativa atrativa para minimizar individualmente o custo do processamento da junção-estrela sobre dados convencionais e o custo do cálculo das operações de distância entre imagens (seção 3). Esses ambientes também proveem disponibilidade dos dados, tolerância a falhas e estratégias para replicação. Na área médica, eles facilitam o compartilhamento de dados e possibilitam que o suporte à tomada de decisão seja mais robusto [Sebaa et al. 2018].

O objetivo deste projeto de mestrado é propor métodos voltados ao processamento eficiente de consultas analíticas estendidas com predicado de similaridade sobre um DW de imagens utilizando o *framework* de processamento paralelo e distribuído Spark [Zaharia et al. 2010]. Embora o trabalho a ser desenvolvido seja genérico, ele tem como motivação a manipulação de imagens médicas, devido à importância da tomada de decisão analítica considerando essas imagens e seu impacto para a sociedade. Portanto, esse contexto é considerado ao longo de todo o artigo.

Este artigo está estruturado da seguinte forma. A fundamentação teórica é descrita na seção 2 e a revisão sistemática é resumida na seção 3. O estado atual do desenvolvi-

mento do projeto e os resultados preliminares são detalhados na seção 4. As considerações finais e as próximas atividades a serem desenvolvidas são listadas na seção 5.

2. Fundamentação Teórica

2.1. *Data warehouse* de imagens

Na Figura 1 é ilustrado um DW de imagens da área médica. Em implementações relacionais, os dados são armazenados segundo o esquema-estrela, com uma tabela de fatos (*Exame*) que se relaciona com várias tabelas de dimensão convencionais (*Paciente, DescriçãoExame, Hospital, DataExame*). Também existem tabelas de dimensão projetadas para armazenar as características intrínsecas das imagens [Teixeira et al. 2015]. A tabela *Vetor Características* armazena os vetores de características de todas as imagens do DW. Esses vetores representam as características das imagens de acordo com as camadas perceptuais definidas no predicado de similaridade. Para cada camada perceptual há um vetor de características obtido por um extrator de características (ex: cor, textura e forma).

A partir dos vetores de características e de uma função de distância, que mede o grau de dissimilaridade entre as imagens, define-se o espaço métrico [Traina et al. 2007]. Esse espaço possibilita a execução de operações de similaridade, como a operação *range query*, a qual, a partir de um centro de consulta, retorna todas as imagens que estão a uma distância menor ou igual ao raio dado. Para diminuir a quantidade de cálculos de distância nas operações de similaridade, a técnica Omni [Traina et al. 2007] define elementos representativos que criam uma região de interesse no espaço métrico, sendo que somente as distâncias às imagens dentro dessa região sejam calculadas. Cada tabela de dimensão $CamadaPerceptual_i$ contém as distâncias entre cada imagem do DW e cada elemento representativo relativo à camada perceptual i.

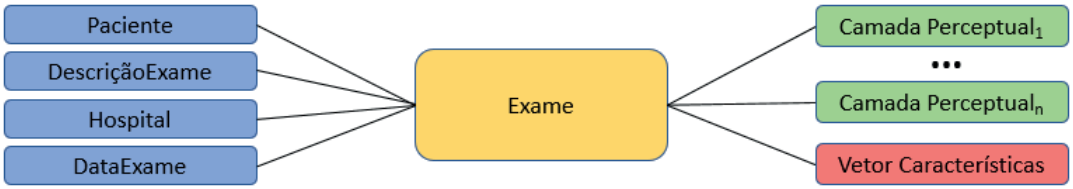


Figura 1. Exemplo de um *data warehouse* de imagens da área médica.

A organização do DW segundo o esquema-estrela requer a realização de operações de junção-estrela no processamento de consultas OLAP. Essas operações realizam junções entre a tabela de fatos e cada uma das tabelas de dimensão envolvidas na consulta, bem como resolvem condições de seleção e agrupamento.

2.2. Ambientes computacionais paralelos e distribuídos

Ambientes computacionais paralelos e distribuídos são baseados no sistema de arquivos distribuído HDFS [Shvachko et al. 2010], que divide o arquivo de dados em blocos, distribuindo e replicando esses blocos em nós do *cluster*. Para abstrair a complexidade inerente ao paralelismo, surgem os *frameworks* MapReduce [Dean and Ghemawat 2008] e Spark [Zaharia et al. 2010]. MapReduce usa um modelo de programação genérico composto de funções *map* e *reduce*, e Spark baseia-se em computação em memória e na abstração de RDD (*resilient distributed dataset*), sendo disponíveis como Apache Hadoop MapReduce (https://hadoop.apache.org/) e Apache Spark (https://spark.apache.org/).

Nesses ambientes, duas técnicas têm sido usadas para melhorar o processamento da junção-estrela [Brito et al. 2016]. A técnica de *broadcast join* assume que as tabelas de dimensão são suficientemente pequenas para serem enviadas para todos os nós durante o processamento da consulta, e realiza todas as junções, em paralelo, localmente em cada nó. Já a técnica de *bloom filter cascade join* usa uma estrutura de dados probabilística para diminuir o tamanho da tabela de fatos antes de realizar a junção-estrela em cascata.

3. Revisão Sistemática

A revisão sistemática analisa o estado-da-arte, identifica lacunas a serem exploradas e detecta tecnologias dentro do contexto da pesquisa [Kitchenham and Charters 2007]. A revisão realizada visou identificar artigos que atendessem às questões: (i) Como realizar junção-estrela em Hadoop? (ii) Como realizar operações de similaridade de imagens em Hadoop? (iii) Como se caracteriza um DW da área médica em Hadoop? (iv) Como realizar operações de junção-estrela e de similaridade sobre um DW em Hadoop? Hadoop foi definido porque mais artigos relacionados à proposta deste projeto foram retornados usando-se essa palavra-chave ao invés de "processamento paralelo e distribuído". Foram consideradas as fontes de busca IEEEXplore Digital Library (https://ieeexplore.ieee.org), Springer (https://www.springer.com/ComputerScience), ACM Digital Library (https://dl.acm.org) e Elsevier (https://www.elsevier.com/physical-sciences/computer-science). Foram analisados os últimos 5 anos, ou seja, de 2014 até o momento.

Na Figura 2 é ilustrada a revisão sistemática realizada, com as palavras-chave, as fontes de busca e as quantidades de artigos retornados. Os idiomas selecionados foram Português e Inglês, sendo as palavras-chave em Inglês omitidas por falta de espaço. Dos 103 artigos retornados, 87 foram excluídos na seleção inicial por meio da leitura do título e resumo, e mais 5 foram excluídos na seleção final por meio da leitura do artigo completo. Os 11 artigos remanescentes foram agrupados de acordo com as questões definidas.

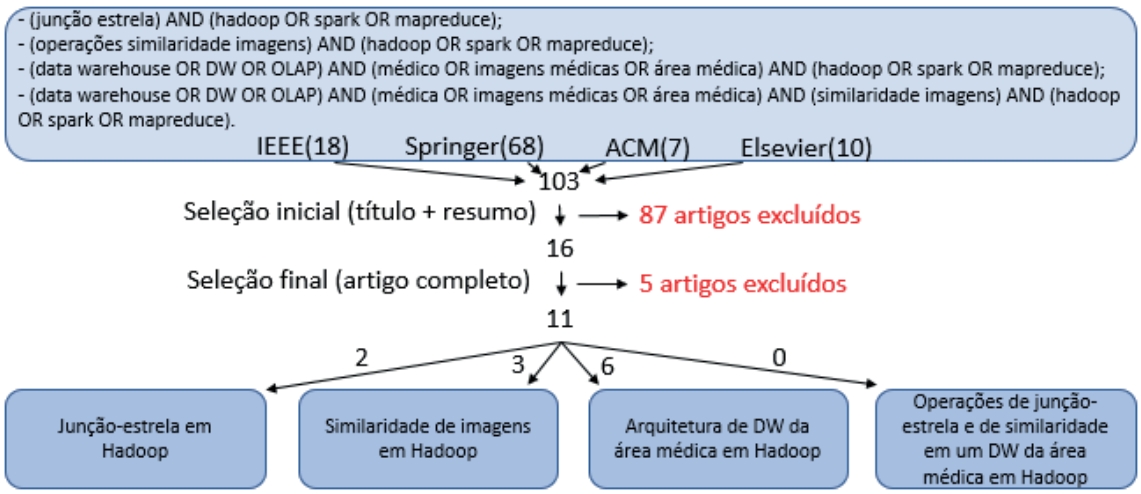


Figura 2. Processo de revisão sistemática que analisou o estado-da-arte.

Os trabalhos de [Guoliang and Guilan 2015, Brito et al. 2016] processam a *junção-estrela* usando MapReduce e Spark. Desses trabalhos, os métodos SBFCJ (*Spark Bloom Filter Cascade Join*) e SBJ (*Spark Broadcast Join*) [Brito et al. 2016] representam

o estado-da-arte, e empregam, em Spark, as técnicas de *bloom filter cascade join* e *broadcast join*, respectivamente. Entretanto, trabalhos desse grupo não manipulam imagens.

Os trabalhos de [Li et al. 2017, Nguyen et al. 2016, Nguyen and Huh 2017] otimizam *operações de similaridade de imagens* em MapReduce e Spark definindo funções *hash* e o uso do método de acesso métrico VP-tree. Limitações incluem a complexidade de se definir funções *hash* apropriadas e o fato de que a Omni tem melhor desempenho que a VP-tree [Traina et al. 2007]. Esses trabalhos não otimizam operações de junção-estrela.

Na proposta de *arquitetura de DW da área médica*, [Istephan and Siadat 2015, Istephan and Siadat 2016, Kuo et al. 2015, Raja and Sivasankar 2014, Sebaa et al. 2017, Sebaa et al. 2018] evidenciaram as vantagens de se usar processamento paralelo e distribuído em aplicações médicas e destacaram as tecnologias Hive e HBase. Porém, esses trabalhos não processam consultas analíticas estendidas com predicado de similaridade.

No melhor do conhecimento dos autores deste artigo, não existem trabalhos que realizem *operações de junção-estrela e de similaridade sobre um DW da área médica* em Hadoop. O projeto de mestrado visa preencher essa lacuna, considerando Spark.

4. Proposta e Estágio Atual de Desenvolvimento

4.1. Descrição da Proposta

Dado o objetivo do projeto de mestrado, estão sendo desenvolvidos dois métodos para o processamento de consultas analíticas estendidas com predicado de similaridade sobre um DW de imagens da área médica em Spark. A descrição a seguir é baseada na Figura 1.

As consultas alvo possuem dois tipos de predicado. O *predicado convencional* é composto por condições de seleção, sendo cada condição definida sobre um atributo de uma tabela de dimensão convencional. O *predicado de similaridade* é composto por uma operação de similaridade e pelas camadas perceptuais consideradas. Na consulta *"Liste a quantidade de imagens similares a uma dada imagem, para pacientes do sexo feminino com câncer de mama e as camadas perceptuais de cor e textura"*, tem-se: (i) condições de seleção: pacientes do sexo feminino e câncer de mama; (ii) operação *range query* para resolver a similaridade; e (iii) camadas perceptuais: cor e textura.

O primeiro método integra as técnicas *broadcast join* e Omni da seguinte forma. Cada k tabela de dimensão convencional envolvida na consulta é filtrada de acordo com as condições de seleção correspondentes, sendo o resultado armazenado na estrutura $HashMapConvencional_k$. Cada i tabela $CamadaPerceptual_i$ envolvida na consulta é filtrada pelo predicado de similaridade usando a Omni, gerando imagens candidatas armazenadas na estrutura $HashMapFiltragem_i$. Por *broadcast*, as estruturas $HashMapFiltragem_i$ são transmitidas e processadas sobre a tabela *Vetor Características* para calcular a distância entre cada imagem candidata e os elementos representativos da camada perceptual correspondente, sendo o resultado armazenado na estrutura $HashMapRefinamento$. Por *broadcast*, as estruturas $HashMapConvencional_k$ e $HashMapRefinamento$ são transmitidas e processadas sobre a tabela de fatos para a realização da junção-estrela. Esse primeiro método é adequado para ambientes nos quais os nós possuem memória primária suficiente para armazenar as tabelas de dimensão, ou seja, eles possuem memória suficiente para processar todas as estruturas *hash map* usadas. Detalhes sobre esse método são descritos em [Rocha and Ciferri 2019].

O segundo método integra as técnicas *bloom filter cascade join* e Omni de forma similar ao primeiro método. Porém, ele usa a estrutura *bloom filter* ao invés da estrutura *hash map* quanto à aplicação dos filtros convencionais e de similaridade, gerando as estruturas $BloomFilterConvencional_k$ e $BloomFilterFiltragem_i$. As estruturas $BloomFilterFiltragem_i$ são processadas sobre a tabela *Vetor Características* gerando a estrutura $BloomFilterRefinamento$. As estruturas $BloomFilterConvencional_k$ e $BloomFilterRefinamento$ atuam sobre a tabela de fatos como filtro, diminuindo o tamanho dessa tabela para o processamento das operações de junção-estrela em cascata. Esse segundo método é adequado para ambientes nos quais as memórias primárias dos nós sejam insuficientes para processar as estruturas completamente.

4.2. Validação da Proposta

Testes de desempenho compararam o primeiro método proposto com o trabalho mais próximo na literatura: SBJ [Brito et al. 2016] (seção 3). O esquema-estrela da Figura 1 foi povoado com dados gerados pela ferramenta ImgDW [Rocha and Ciferri 2018], que já representa uma contribuição do trabalho de mestrado. Foram definidas consultas analíticas com diferentes condições de seleção e aplicada a operação *range query* (raio de abrangência de 20% do diâmetro do conjunto de dados) para calcular a similaridade, considerando as camadas perceptuais *Haralick Variância* (baixa dimensionalidade: 4 dimensões) e *Histograma de Cores* (alta dimensionalidade: 256 dimensões). Foi usado um *cluster* com 5 nós, cada qual com no mínimo 3GB de RAM. Os resultados mostraram que, para consultas contendo pelo menos a camada perceptual de alta dimensionalidade, o método proposto proveu ganhos de desempenho que variaram de 60,01% a 65,49%. Para consultas contendo apenas a camada perceptual de baixa dimensionalidade, o método proposto empatou ou proveu ganhos de desempenho de até 10,50%. A variação do desempenho em termos da dimensionalidade das camadas perceptuais é uma característica herdada da técnica Omni [Traina et al. 2007]. Detalhes sobre os resultados obtidos são descritos em [Rocha and Ciferri 2019].

5. Conclusão

No projeto de mestrado estão sendo desenvolvidos dois métodos para o processamento de consultas analíticas com predicado de similaridade sobre um DW de imagens da área médica em Spark. O primeiro método integra as técnicas de *broadcast join* e Omni e o segundo método integra as técnicas de *bloom filter cascade join* e Omni. As próximas atividades referem-se à continuidade do processo de validação do primeiro método, com a definição de um ambiente de teste mais robusto considerando novas consultas e testes de escalabilidade, bem como a implementação e validação do segundo método proposto.

Referências

Brito, J. J., Mosqueiro, T., Ciferri, R. R., and Ciferri, C. D. A. (2016). Faster cloud star joins with reduced disk spill and network communication. In *ICCS 2016*, volume 80 of *Procedia Computer Science*, pages 74–85.

Dean, J. and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Guoliang, Z. and Guilan, W. (2015). GBFSJ: Bloom filter star join algorithms on GPUs. In *FSKD 2015*, pages 2427–2431.

Istephan, S. and Siadat, M.-R. (2015). Extensible query framework for unstructured medical data – a big data approach. In *IEEE ICDMW 2015*, pages 455–462.

Istephan, S. and Siadat, M.-R. (2016). Unstructured medical image query using big data – an epilepsy case study. *Journal of Biomedical Informatics*, 59:218–226.

Kimball, R. and Ross, M. (2002). *The data warehouse toolkit: the complete guide to dimensional modeling, 2nd Edition*. Wiley.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.

Kuo, M., Chrimes, D., Moa, B., and Hu, W. (2015). Design and construction of a big data analytics framework for health applications. In *IEEE SmartCity 2015*, pages 631–636.

Li, D., Zhang, W., Shen, S., and Zhang, Y. (2017). SES-LSH: Shuffle-efficient locality sensitive hashing for distributed similarity search. In *ICWS 2017*, pages 822–827.

Nguyen, D.-T., Yong, C. H., Pham, X.-Q., Nguyen, H.-Q., Loan, T. T. K., and Huh, E.-N. (2016). An index scheme for similarity search on cloud computing using MapReduce over docker container. In *IMCOM 2016*, pages 60:1–60:6.

Nguyen, T. D. T. and Huh, E.-N. (2017). An efficient similar image search framework for large-scale data on cloud. In *IMCOM 2017*, pages 65:1–65:8.

Raja, P. V. and Sivasankar, E. (2014). Modern framework for distributed healthcare data analytics based on hadoop. In *ICT-EurAsia 2014*, pages 348–355.

Rocha, G. M. and Ciferri, C. D. A. (2018). ImgDW generator: a tool for generating data for medical image data warehouses. In *SBBD 2018 Proc. Companion*, pages 23–28.

Rocha, G. M. and Ciferri, C. D. A. (2019). Processamento eficiente de consultas analíticas estendidas com predicado de similaridade em Spark. In *SBBD 2019*, pages 1–6.

Sebaa, A., Chikh, F., Nouicer, A., and Tari, A. (2018). Medical big data warehouse: Architecture and system design, a case study: Improving healthcare resources distribution. *Journal of Medical Systems*, 42(4):59.

Sebaa, A., Nouicer, A., Chikh, F., and Tari, A. (2017). Big data technologies to improve medical data warehousing. In *BDCA 2017*, pages 21:1–21:5.

Shvachko, K., Kuang, H., Radia, S., and Chansler, R. (2010). The Hadoop distributed file system. In *IEEE MSST 2010*, pages 1–10.

Teixeira, J. W., Annibal, L. P., Felipe, J. C., Ciferri, R. R., and Ciferri, C. D. A. (2015). A similarity-based data warehousing environment for medical images. *Computers in Biology and Medicine*, 66:190 – 208.

Traina, C., Filho, R. F. S., Traina, A. J. M., Vieira, M. R., and Faloutsos, C. (2007). The omni-family of all-purpose access methods: a simple and effective way to make similarity search more efficient. *The VLDB Journal*, 16(4):483–505.

Traina, C., Moriyama, A., Rocha, G. M., Cordeiro, R., Ciferri, C. D. A., and Traina, A. J. M. (2019). The SimilarQL framework: similarity queries in plain SQL. In *SAC 2019*, pages 1–4.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: Cluster computing with working sets. In *USENIX HotCloud 2010*.

FeSHyD: Busca Federada sobre Bases de Dados RDF Híbridas

Hugo Paulino Bonfim Takiuchi[1]
Orientadora: Carmem Satie Hara[1]
Coorientadora: Raqueline Ritter de Moura Penteado[2]

[1] Departamento de Informática – Universidade Federal do Paraná
Caixa Postal 19.081 – 81.531-990 – Curitiba, PR – Brasil

[2]Departamento de Informática – Universidade Estadual de Maringá
Avenida Colombo 5.790 – 87.020-900 – Maringá, PR – Brasil
{hpbtakiuchi,carmem}@inf.ufpr.br, raque@din.uem.br

Nível: Mestrado
Ingresso no Programa: Março/2018
Previsão de Conclusão: Março/2020
Etapas Concluídas: Conclusão dos créditos, defesa da proposta Agosto/2019.
Etapas Futuras: Implementação, análise, defesa da dissertação

Resumo. *Na Web Semântica, os dados são disponibilizados no formato RDF e consultados por meio da linguagem SPARQL. A maioria dos processadores de consultas consideram apenas bases RDF de terceiros ou apenas uma base proprietária. Bases de terceiros consistem de um conjunto de repositórios autônomos, enquanto bases proprietárias permitem acesso irrestrito, tanto aos dados quanto ao processamento interno da consulta. Nesta dissertação é proposto um framework denominado FeSHyD, que processa consultas SPARQL tanto sobre bases de terceiros quanto uma base proprietária distribuída. O FeSHyD divide a consulta para ser processada individualmente por cada fonte de dados autônoma, bem como gera um plano de consulta otimizado para ser executado pela base proprietária. Durante a geração do plano, a otimização envolve métodos para seleção das fontes, bem como a ordenação das subconsultas. Durante o processamento da consulta, o framework permite que os servidores que compõem a base proprietária submetam consultas às bases de terceiros diretamente, sem a existência de um ponto central de controle.*

Palavras-chave: *busca federada, consulta SPARQL, bases de dados distribuídas, integração de sistemas distribuídos*

1. Introdução

Parte dos dados publicados na Internet atualmente segue os padrões da Web Semântica[1], que é uma extensão da Web de documentos. Os objetivos da Web Semântica são produzir documentos legíveis à máquina visando a automação na análise dos recursos e conseguir integrar dados dispersos em vários sites [Hendler 2001]. Essa integração leva em consideração o conceito de dados ligados. São criadas ligações entre informações de diferentes locais com a finalidade de facilitar a busca por informações relevantes. Os dados na Web Semântica são dispostos no formato RDF (*Resource Description Framework*) e acessados via a linguagem de consulta SPARQL *(SPARQL Protocol and RDF Query Language)*.

Uma base de dados RDF consiste de um conjunto de triplas. Atualmente algumas bases de dados apresentam um grande volume de dados e comportam cerca de 1 trilhão de triplas[2]. Com a finalidade de dar suporte a este grande volume, alguns trabalhos propõem a utilização do armazenamento de dados RDF distribuído em diferentes servidores. Uma abordagem para acessar os dados distribuídos é conhecido como busca federada, na qual um moderador é responsável por coordenar o processamento e execução da consulta sobre bases de dados autônomas [Rakhmawati et al. 2013]. As bases autônomas podem pertencer a terceiros, e neste caso o moderador só tem acesso à base através de consultas SPARQL, ou as bases podem ser de propriedade da própria aplicação. As bases proprietárias permitem otimizações durante a execução interna da consulta. Ou seja, bases de terceiros são acessadas como caixas-pretas, enquanto bases proprietárias podem ser acessadas como caixas-brancas.

Com o intuito de explorar possíveis otimizações de bases proprietárias distribuídas, esta dissertação propõe o framework FeSHyD. Este sistema realiza uma busca federada, na qual a consulta SPARQL é processada e executada sobre uma combinação de bases de terceiros com uma base de dados proprietária distribuída. Como forma de acesso a uma base proprietária é proposto utilizar o sistema PAbS, que particiona os dados em servidores dentro de um cluster [Penteado et al. 2019]. Dentre as otimizações propostas é planejado permitir que os servidores que compõem a base proprietária enviem requisições às bases de terceiros diretamente, sem a interferência do moderador. O objetivo do sistema FeSHyD é diminuir o tempo de execução e melhorar o desempenho da consulta através da descentralização e distribuição das atividades envolvidas no processamento sobre as bases de terceiros. Este artigo está dividido da seguinte forma: na Seção 2 são apresentados alguns conceitos básicos. Na Seção 3 são descritos brevemente os trabalhos relacionados. Na Seção 4 é apresentado o modelo de busca federada proposto e, por fim, na Seção 5 são descritas as considerações finais.

2. Conceitos Básicos

Uma base de dados RDF é formada de triplas *(sujeito, predicado, objeto)*, que podem ser visualizadas como um grafo direcionado (*grafo de dados*), no qual sujeitos e objetos são representados por vértices, ligados através de arestas que representam predicados[3]. O

[1] `https://www.w3.org/standards/semanticweb/`
[2] `https://www.w3.org/wiki/LargeTripleStores`
[3] `https://www.w3.org/TR/rdf11-primer/`

RDF Schema (RDFS)[4] é uma extensão semântica do modelo RDF, a qual define recursos como propriedades, classes e instâncias. Com o auxílio do RDFS é possível elaborar o esquema estrutural que descreve como os dados estão relacionados (*grafos de estruturas*). Para realizar o acesso aos dados RDF a linguagem de consulta SPARQL foi proposta pelo consórcio W3C [5]. Uma busca SPARQL é formada por padrões de triplas, que reunidas formam o *grafo da consulta*. Os padrões de triplas podem conter variáveis em cada uma das suas posições, que são mapeadas para valores da base de dados durante a execução da consulta. A Figura 1 (a) ilustra um exemplo de consulta SPARQL contendo três padrões de triplas. Esta consulta obtém o nome e valor de oferta de produtos, cujo valor ofertado seja menor que 1500. O grafo de estrutura desta base é apresentado na Figura 1 (c).

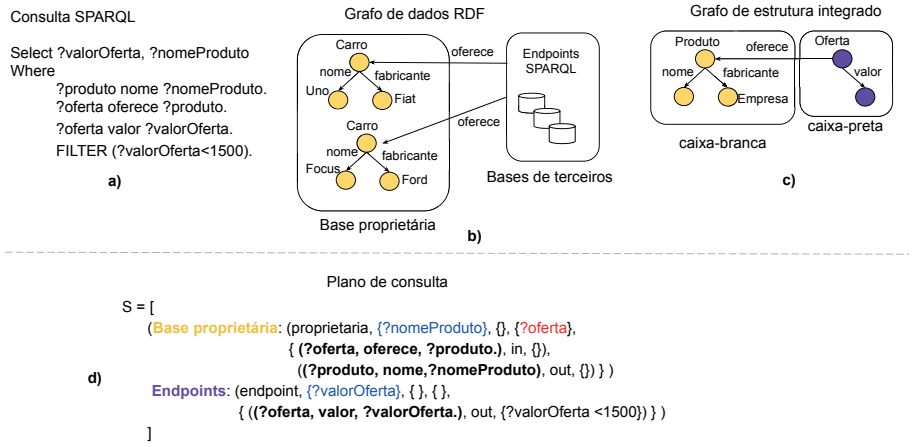


Figura 1. a) Consulta SPARQL. b) Grafo de dados. c) Grafo do esquema estrutural. d) Plano de consulta criado pelo framework FeSHyD

Com relação ao armazenamento, os dados podem estar organizados de maneira centralizada ou distribuída [Rakhmawati et al. 2013]. Em uma base com armazenamento *centralizado* os dados estão localizados em apenas uma máquina, enquanto que bases *distribuídas* particionam os dados entre várias servidores. Neste caso, há a figura do *moderador*, que recebe requisições sobre a base distribuída e coordena a busca e processamento entre os diversos servidores. Com relação à propriedade, as bases de dados pertencem ao mesmo *proprietário* do sistema de consulta ou os dados estão alocados em bases *pertencentes à terceiros*. A Figura 1 (b) mostra uma base de dados composta de uma base proprietária contendo as informações relativas ao produto e bases externas que armazenam as ofertas. Nas bases de terceiros, o serviço responsável por receber requisições de consulta e retornar resultados é conhecido como *endpoint*. Com relação ao processamento da consulta, ele é considerado *local* quando o moderador e os dados estão na mesma máquina; caso contrário, o processamento é considerado *remoto*. As bases de dados podem apresentar acesso restrito tanto aos dados quanto sobre o processamento interno da consulta. Bases restritivas são acessadas como *caixas-pretas* e as bases que permitem modificação nos dados e na execução da consulta são acessadas como *caixas-brancas*. As bases de dados caixas-pretas podem ser processadas com *links transversais*, que descobre as bases de terceiros em tempo de execução. A descoberta de novas bases acontece percorrendo *links* entre bases encontrados durante o processamento da consulta.

[4] https://www.w3.org/TR/rdf-schema/
[5] https://www.w3.org/TR/rdf-sparql-query/

Durante o percurso são avaliadas quais as fontes que respondem os padrões de triplas.

Outro tipo de processamento sobre bases de dados consideradas caixas-pretas é a *busca federada*, que utiliza o conhecimento prévio das bases na criação e execução do plano de consulta. Um exemplo de busca federada sobre dados RDF é apresentado na Figura 2. A fase de planejamento compreende as seguintes etapas: análise da consulta SPARQL, seleção de fontes, decomposição da consulta original em subconsultas e ordenação das mesmas. Ao fim da fase de planejamento, são criados diversos planos de consulta. A fase de execução da consulta processa o plano com menor custo criado durante o planejamento. Nesta fase é realizada a etapa de escolha do método de junção dos resultados intermediários [Rakhmawati et al. 2013].

3. Trabalhos Relacionados

Vários sistemas foram criados com a finalidade de realizar buscas sobre bases de dados RDF [Rakhmawati et al. 2013]. Alguns destes sistemas são baseados em consultas federadas sobre bases de dados autônomas. oLinDa apresenta uma técnica de decomposição de consultas baseada em esquemas estruturais [da Cunha and Lóscio 2014]. O SPLENDID utiliza metadados sobre as bases e consultas SPARQL na seleção de fontes [Görlitz and Staab 2011]. No Lusail as variáveis de junção entre padrões de triplas determinam quais as bases de dados relevantes [Abdelaziz et al. 2017]. O sistema FedX [Schwarte et al. 2011] opera sobre bases de terceiros distribuídas (endpoints remotos) e centralizadas (copia local das bases), utilizando consultas SPARQL na seleção de fontes. Além de bases autônomas, o sistema Ephedra [Nikolov et al. 2017] processa uma busca sobre uma base proprietária distribuída vista como caixa-preta. Modelos de dados diferentes e interfaces de consultas variadas são consideradas no processamento da busca neste sistema. Diferente dos outros sistemas citados, o sistema SIHJoin [Ladwig and Tran 2011] divide a consulta entre bases de terceiros distribuídas e uma base proprietária centralizada vista como caixa-branca. SIHJoin é baseado em links transversais no acesso aos endpoints externos.

A Tabela 1 apresenta uma comparação entre o tipo de armazenamento e processamento dos dados nos sistemas descritos nesta seção com o framework FeSHyD, proposto nesta dissertação. Diferente do SIHJoin, o FeSHyD aborda o armazenamento híbrido considerando uma base de dados proprietária distribuída e realiza o processamento da consulta sobre os endpoints externos através de busca federada.

	Armazenamento		**Processamento**	
Critérios / **Sistemas**	**Bases Proprietárias**	**Bases de Terceiros**	**Localidade**	**Formas de Acesso**
SPLENDID	X	Distribuída	Remoto	Caixa-Preta (busca federada)
oLinDa	X	Distribuída	Remoto	Caixa-Preta (busca federada)
Lusail	X	Distribuída	Remoto	Caixa-Preta (busca federada)
Ephedra	Distribuída	Distribuída	Remoto	Caixa-Preta (busca federada)
FedX	X	Centralizada e Distribuída	Local e Remoto	Caixa-Preta (busca federada)
SIHJoin	Centralizada	Distribuída	Local e Remoto	Caixa-Branca e Caixa-Preta (links transversais)
FeSHyD	Distribuída	Distribuída	Remoto	Caixa-Branca e Caixa-Preta (busca federada)

Tabela 1. Comparação entre os sistemas de consulta distribuída

4. Modelo Proposto

A proposta deste trabalho é criar o framework denominado FeSHyD *(**Fe**derated **S**earch on **Hy**brid **D**atabases)*, que realiza consultas federadas SPARQL otimizadas, considerando um modelo de armazenamento contendo bases de dados RDF híbridas. Estas bases RDF são consideradas híbridas pois são compostas de bases de terceiros acessadas como caixas-pretas e uma base proprietária distribuída vista como caixa-branca. O objetivo é elaborar e implementar otimizações considerando o conhecimento e acesso irrestrito, tanto aos dados quanto ao planejamento e execução da consulta, sobre a base de dados proprietária distribuída. O sistema visa a descentralização do processamento da consulta realizado pelo moderador, dividindo parte da responsabilidade para a base de dados proprietária. Com isso, é esperado diminuir o tempo de execução total da consulta. A Figura 2 apresenta a arquitetura proposta de busca federada considerando uma composição híbrida de armazenamento com endpoints SPARQL e uma base proprietária distribuída. Nesta figura o moderador analisa e executa a consulta, como descrito na Seção 2.

A base para o desenvolvimento do framework FeSHyD é o gerenciador de bases RDF distribuídas PAbS [Penteado et al. 2019]. O PAbS particiona a base de dados a partir de padrões de alocação (esquemas estruturais) e utiliza estas informações de co-alocação de dados para gerar planos de consultas otimizados. Um exemplo de plano de consulta gerado pelo PAbS é apresentado nas primeiras 3 linhas da Figura 1 (d). O plano de consulta é dividido em blocos, onde cada bloco contém uma sequência de passos com: o tipo da base de dados destino, as variáveis buscadas, as variáveis de ligação entre os padrões de tripla, o padrão de tripla, a direção de exploração do grafo da consulta e os filtros aplicados.

Contudo, o PAbS não permite acesso a bases externas caixas-pretas. Assim, para processar a consulta ilustrada na Figura 1(a), que é obter o nome de produtos e valor de ofertas, no contexto ilustrado na Figura 1(b), em que as ofertas estão armazenadas em bases de terceiros, há 2 alternativas: (i) utilizar um sistema de consultas federado e considerar a base proprietária do PAbS como uma caixa-preta; (ii) ter uma intervenção do usuário para obter resultados intermediários para encaminhar às bases de terceiros. O FeSHyD oferece uma terceira alternativa quando há conhecimento dos grafos de estrutura das bases de terceiros: estender a geração de planos de consulta do PAbS com acessos a estas bases, como ilustrado no plano da Figura 1 (d). Além de facilitar o acesso a bases híbridas esta alternativa apresenta novas possibilidades de otimização no processamento da consulta. Como no PAbS todos os servidores de armazenamento recebem o mesmo plano de consulta e iniciam o seu processamento em paralelo, é possível também paralelizar o acesso às bases de terceiros diretamente por estes servidores, sem a necessidade de intervenção do moderador. Ou seja, é possível diminuir a carga do moderador por meio da paralelização de requisições de subconsultas SPARQL para os endpoints caixas-pretas a partir dos servidores de armazenamento diretamente. As otimizações planejadas para serem exploradas pelo FeSHyD são detalhadas na próxima seção.

4.1. Otimizações Propostas

Na dissertação serão exploradas duas otimizações, uma na fase de planejamento e outra na fase de execução. Durante o planejamento, será considerada a seletividade das subconsultas (blocos) que compõem o plano de execução para que a exploração de grafos

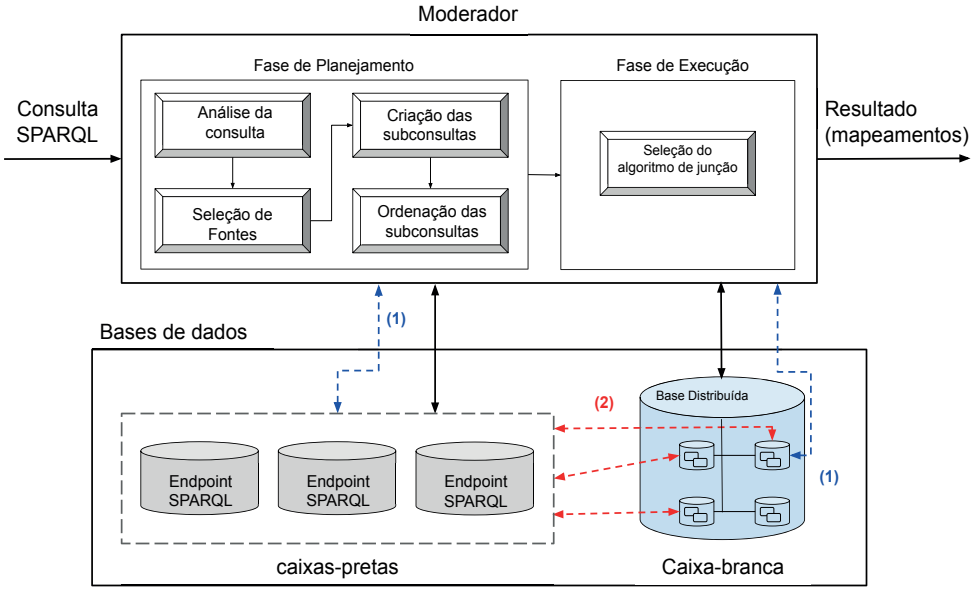


Figura 2. Arquitetura de busca federada híbrida, acessando endpoints SPARQL e uma base proprietária distribuída

inicie pelas mais seletivas, a fim de minimizar a quantidade de resultados intermediários. Quando a consulta envolve dados tanto da base proprietária como de terceiros, há diversas alternativas a serem exploradas durante a fase de execução. Na primeira, o moderador possui controle total do processamento, o que se assemelha a buscas federadas tradicionais e indicada com a linha tracejada (1) na Figura 2. Na segunda, chamada de *requisição direta* e indicada com a linha tracejada (2), as requisições aos endpoints SPARQL são feitas diretamente pelos servidores de armazenamento da base proprietária, sem intermediação do moderador. Neste caso, os servidores devem ser capazes de criar consultas SPARQL e se comunicar com os endpoints remotos. Servidores que realizam as funções tanto de armazenamento como de comunicação são denominados de servidores híbridos.

4.2. Estado Atual e Atividades

As etapas já concluídas da dissertação são listadas abaixo.

- Levantamento bibliográfico
- Elaboração e defesa da proposta
- Análise da consulta SPARQL: Identificar e representar o grafo da consulta.
- Desenvolvimento da interface entre sistemas: Definir a comunicação entre o FeSHyD e o sistema PAbS.

Além da proposta da dissertação, como resultado destas etapas, foi submetido um artigo demo para o SBBD 2019 [Penteado et al. 2019]. O cronograma das etapas futuras está apresentado na Tabela 2. Para a validação do sistema, planeja-se utilizar o benchmark Berlin e considerar como *baseline* de comparação um sistema de buscas federadas tradicionais, que trata a base proprietária como uma caixa-preta. Serão considerados o tempo de resposta da consulta e vazão como critérios de avaliação. Uma variação que pode ser considerada é a quantidade crescente de bases de terceiros utilizadas no framework.

5. Considerações Finais

Este trabalho apresentou o sistema de busca federada FeSHyD, que realiza consultas sobre bases de dados RDF híbridas. Neste framework é proposto utilizar uma base de dados

	2019					2020		
Atividade	08	09	10	11	12	1	2	3
Defesa da proposta	X							
Seleção de algoritmos	X	X						
Implementação da geração de consulta		X	X					
Implementação da comunicação com endpoints			X	X				
Validação com experimentos				X	X	X		
Escrita de artigo							X	X
Escrita da dissertação				X	X	X	X	

Tabela 2. Cronograma de atividades

distribuída proprietária com a finalidade de otimizar a consulta internamente, bem como bases de terceiros acessadas como caixas-pretas. O framework proposto utiliza o sistema PAbS, que particiona os dados RDF entre servidores distribuídos em um cluster. É proposto delegar parte das responsabilidades do moderador a estes servidores, descentralizando o processamento da consulta e diminuindo a sobrecarga no mesmo. Como resultado, é esperado que o sistema FeSHyD otimize o tempo de execução e desempenho de uma busca federada SPARQL.

Referências

Abdelaziz, I., Mansour, E., Ouzzani, M., Aboulnaga, A., and Kalnis, P. (2017). Lusail: a system for querying linked data at scale. *Proceedings of the VLDB Endowment*, 11(4):485–498.

da Cunha, D. R. B. and Lóscio, B. F. (2014). oLinDa: uma abordagem para decomposição de consultas em federações de dados interligados. In *Anais do Simpósio Brasileiro de Banco de Dados*, pages 137–146.

Görlitz, O. and Staab, S. (2011). Splendid: SPARQL endpoint federation exploiting VoID descriptions. In *Proceedings of the Second International Conference on Consuming Linked Data-Volume 782*, pages 13–24. CEUR-WS. org.

Hendler, J. (2001). Agents and the semantic Web. *IEEE Intelligent Systems*, 16(2):30–37.

Ladwig, G. and Tran, T. (2011). SIHJoin: querying remote and local linked data. In *Extended Semantic Web Conference*, pages 139–153. Springer.

Nikolov, A., Haase, P., Trame, J., and Kozlov, A. (2017). Ephedra: Efficiently combining RDF data and services using SPARQL federation. In *International Conference on Knowledge Engineering and the Semantic Web*, pages 246–262. Springer.

Penteado, R. R. M., Takiuchi, H. P. B., and Hara, C. S. (2019). PAbS: Um processador de consultas SPARQL sobre bases distribuídas. In *submetido para o SBBD Demo*.

Rakhmawati, N. A., Umbrich, J., Karnstedt, M., Hasnain, A., and Hausenblas, M. (2013). Querying over federated SPARQL endpoints—a state of the art survey. *arXiv preprint arXiv:1306.1723*.

Schwarte, A., Haase, P., Hose, K., Schenkel, R., and Schmidt, M. (2011). Fedx: Optimization techniques for federated query processing on linked data. In *International Semantic Web Conference*, pages 601–616. Springer.

Geração de Dados ECG Sintéticos usando Redes Gerativas Adversárias (GAN)

Cristiano Sousa Melo[1], **José Maria da Silva Monteiro Filho**[1]

[1]Programa de Mestrado e Doutorado em Ciencia da Computação (MDCC)
Universidade Federal do Ceará (UFC)
Campus do Pici – Centro de Ciências – Bloco 910 – 60.455-760 – Fortaleza – CE – Brazil

cristianomelo_88@hotmail.com, monteiro@dc.ufc.br

Nível: Mestrado
Ingresso: Fevereiro de 2018
Previsão de Término: Fevereiro de 2020
Etapas já concluídas: Revisão Bibliográfica Preliminar, Definição do Problema, Defesa de Qualificação
Defesa da Pré-Proposta: Novembro de 2019
Defesa da Proposta: Janeiro de 2020

Abstract. *Chagas disease is an infection caused by the Trypanosoma cruzi protozoan, which can present an acute or chronic phase. The latter, symptomatic (determined), affects the nervous, digestive, and cardiac systems. About two-thirds of people with chronic symptoms shows alterations in heart function which causes abnormal heart rhythms and can result in sudden death. Using electrocardiogram (ECG), a low-cost signal data, prediction models can be used to predict the sudden death of these patients. In general, such models require a large amount of data to generate good classifiers. However, there are few ECG signs available on patients with Chagas disease who have had sudden death. In this context, this work aims to generate data of synthetic ECG signals, with the characteristics of these patients, through the use of Generative Adversarial Nets (GAN) in order to improve the accuracy of the models.*
Keywords: Generative Adversarial Net, Recurrent Neural Network, Deep Learning

Resumo. *A doença de Chagas é uma infecção causada pelo protozoário Trypanosoma cruzi, que pode apresentar uma fase aguda ou crônica. Este último, sintomático (determinado), afeta os sistemas nervoso, digestivo e cardíaco. Cerca de dois terços das pessoas com sintomas crônicos mostram alterações na função cardíaca que causam ritmos cardíacos anormais e podem resultar em morte súbita. Utilizando o eletrocardiograma (ECG), um dado de baixo custo, modelos de predição podem ser usados para prever a morte súbita desses pacientes. Em geral, esses modelos requerem uma grande quantidade de dados para gerar bons classificadores. No entanto, existem poucos sinais de ECG disponíveis em pacientes com doença de Chagas que tiveram morte súbita. Neste contexto, este trabalho tem como objetivo gerar dados de sinais sintéticos de ECG, com as características desses pacientes, através do uso de Redes Gerativas Adversária (GAN) a fim de melhorar a precisão dos modelos.*

1. Introdução

A doença de Chagas é uma infecção causada pelo protozoário *Trypanosoma cruzi*, a qual pode apresentar uma fase aguda e uma fase crônica. A fase crônica, sintomática (determinada), afeta os sistemas nervoso, digestório e cardíaco. Cerca de dois terços das pessoas com sintomas crônicos apresentam alterações no funcionamento do coração, incluindo a miocardiopatia dilatada, que causa anormalidades no ritmo cardíaco e pode resultar em morte súbita [Rassi-Junior et al. 2000].

A Eletrocardiografia é uma técnica utilizada para registrar as alterações de potencial elétrico produzidas pela atividade cardíaca. A evolução temporal das referidas alterações é chamada de sinal eletrocardiograma (ECG) [Geselowitz 1989]. O traçado obtido pelo registro eletrocardiográfico contém uma série de formas de onda e complexos, que foram denominados onda P, complexo QRS e onda T. As ondas ou deflexões são separadas por intervalos regulares. Assim, a despolarização atrial produz a onda P; a despolarização dos ventrículos produz o complexo QRS, e a repolarização ventricular produz a onda T. A análise do comportamento do sinal ECG permite extrair informações diversificadas, as quais podem subsidiar a identificação de uma grande variedade de doenças e anomalias cardíacas [Gonçalves et al. 2007]. A Figura 1 ilustra as ondas características do sinal ECG.

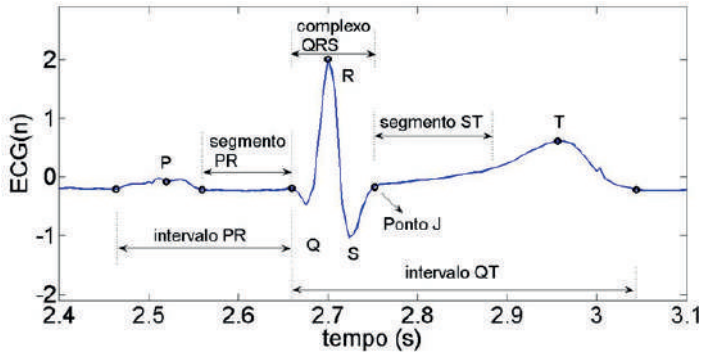


Figura 1. Ondas características e parâmetros de análise do ECG

Neste contexto, algoritmos de mineração de dados e redes neurais profundas têm sido utilizados com sucesso na classificação de batimentos cardíacos e detecção de arritmias [Rahhal et al. 2016], [Dalvi et al. 2016], [Luz et al. 2016], [Ramakrishnan et al. 2017] e [Xiang et al. 2018]. Um fator fundamental para o êxito desses algoritmos foi a disponibilização de grandes quantidades de sinais ECG em repositórios públicos, tais como o AHA/MIT-BIH (Physionet) [Goldberger et al. 2000], o SCP-ECG [Mandellos et al. 2010] e HL7 aECG [Bond et al. 2011].

Porém, quando se deseja utilizar algoritmos de aprendizado automático com a finalidade de investigar o comprometimento cardíaco provocado por doenças raras ou de baixa incidência, tais como Amiloidose ou doença de Chagas, esbarra-se na dificuldade de se obter uma base significativa de sinais ECG com as características desejadas. Por exemplo, modelos de predição podem ser utilizados para prever a morte súbita de pacientes com doença de Chagas. Todavia, existem poucos sinais ECG disponíveis sobre pacientes com doença de Chagas que tiveram morte súbita. Neste caso, uma possível solução consiste na geração de dados sintéticos de sinais ECG.

Este trabalho tem por objetivo gerar dados de sinais ECG sintéticos, com as características de pacientes que tiveram morte súbita, através do uso de Redes Gerativas Adversárias (GAN). Espera-se que a utilização dos dados sintéticos gerados possa aumentar a acurácia dos modelos de classificação utilizados para prever os pacientes com doença de Chagas que apresentam risco de morte súbita.

Com isso, os objetivos deste trabalho são caracterizados por:

- Geração de sinais de ECG sintético de paciente chagásicos que tiveram morte súbita utilizando Redes Gerativas Adversárias (GAN);

- Com os dados sintéticos gerados, validá-los com classificadores de doença chagásica a fim de comparar seus resultados com dados reais.

2. Trabalhos Relacionados

A utilização de sinais de ECG em classificadores para detecção de anomalias é uma abordagem que já vem sendo utilizada a bastante tempo. [Souza et al. 2006], por exemplo, fizeram um classificador para detecção de isquemia utilizando uma base de dados com apenas 78 registros. Os *surveys* [Nanarkar and Chawan 2018] e [Celin and Vasanth 2017] fizeram um levantamento dos principais trabalhos feitos sobre detecção de arritmias utilizando algoritmos de classificação de Machine Learning.

Dada a dificuldade de obtenção de registros de sinais ECG para gerar bons classificadores, [Mcsharry et al. 2003] propuseram um gerador de sinais ECG baseado na utilização de técnicas de equações diferenciais. Entretando, o fato do sinal ter sido gerado a partir de uma equação matemática gerou discussões sobre a sua qualidade e generalidade.

Recentemente, as Redes Gerativas Adversárias começaram a ser utilizadas com a finalidade de gerar sinais biomédicos. [Esteban et al. 2018] propuseram dois modelos de GAN: GAN Recorrente (RGAN) e GAN Condicional Recorrente (RCGAN) para produzir série temporal multidimensionais que representam sinais biomédicos, tais como saturação de oxigênio, frequência cardíaca, taxa respiratória e pressão arterial média. Os autores utilizaram o conjunto de dados MNIST e um conjunto de dados chamado eICU.

[Li et al. 2019] propõem um método de detecção de anomalia multivariada não supervisionada baseado em Redes Generais Adversariais (GAN), usando as Redes Neurais Recorrentes de Memória de Curto Prazo (LSTM-RNN) como modelos base (isto é, o gerador e o discriminador) na estrutura da GAN para capturar a correlação temporal de distribuições em série temporal. Em vez de tratar cada fluxo de dados de forma independente, a estrutura proposta de detecção de anomalia multivariada considera a variável inteira definida simultaneamente para capturar as interações latentes entre as variáveis.

Em [Brophy et al. 2019], os autores converteram sinais ECG em imagens de tamanho 64x64, capturando apenas um ciclo de batimento. Em seguida, uma rede GAN, baseada na rquitetura de Wasserstein [Arjovsky et al. 2017], é utilizada para gera imagens de sinais ECG sintéticos. Por fim, as imagens geradas são convertidas em sinais ECG sintéticos e utilizadas em um classificador. Embora já seja possível gerar dados ECG sintéticos, ainda há limitações quanto à qualidade dos sinais gerados. Os sinais gerados ainda possuem um nível elevado de ruído, o que compromete a acurácia dos classificadores.

Neste contexto, o presente trabalho tem por objetivo gerar dados de sinais ECG

sintéticos, com as características de pacientes que tiveram morte súbita, através do uso de Redes Gerativas Adversárias (GAN). Na arquitetura da GAN, pretendemos utilizar redes LSTM tanto em seu gerador quanto em seu discriminador. Desta forma, esperamos que a utilização dos sinais sintéticos gerados possa aumentar a acurácia dos modelos de classificação utilizados para prever os pacientes com doença de Chagas que apresentam risco de morte súbita.

3. Fundamentação Teórica

A seguir, serão descritos os principais conceitos e o conjunto de dados utilizados neste trabalho.

3.1. Redes Gerativas Adversárias (GAN)

As Redes Gerativas Adversárias (GAN) [Goodfellow et al. 2014] são modelos de aprendizagem não supervisionada utilizados para gerar instâncias sintéticas a partir da distribuição de um conjunto de dados de entrada. Essas redes são baseadas na teoria dos jogos [Osborne and Rubinstein 1994], um campo da matemática aplicada que modela situações de interações estratégicas, isto é, quando as decisões de um agente influenciam nas recompensas e nas ações de outros agentes.

A arquitetura de uma GAN é composta por uma rede geradora, que tenta produzir amostras que se assemelhem aos dados originais, $x \sim P_{dados}$, de acordo com uma série de transformações que pode ser descrita por uma função $x = g(z; \theta_g)$. A rede discriminadora, por sua vez, produz a probabilidade de x ser falso ou real, que é dada por uma função $d(x; \theta_d)$. Assim, dado dois jogadores (discriminador e gerador), o problema de aprendizado da GAN consiste como jogo da soma zero [Nash 1950], no qual uma função $r(\theta_d; \theta_g)$ determina a recompensa (*pay-off*) de uma das redes e $-r(\theta_d; \theta_g)$, da outra. Logo, no equilíbrio da soma zero, temos:

$$g^* = arg \min_g \max_d r(g, d)$$

onde g^* é a rede geradora no ponto de convergência, capturando de forma ótima a distribuição dos dados.

3.2. Série Temporal

Uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo. Em séries temporais a ordem das dos dados é de fundamental importância, uma vez que as observações vizinhas são dependentes e o interesse consiste em analisar e modelar essa dependência. Neste contexto, um sinal ECG pode ser representado por uma série temporal.

4. A Solução Proposta

A seguir, serão descritas a arquitetura e o conjunto de dados utilizados neste trabalho.

4.1. Arquitetura Proposta

A Figura 2 mostra uma arquitetura geral da Rede Gerativa Adversária que pretendemos utilizar neste trabalho. Como entrada do discriminador, temos instâncias de janela de tempo de sinais ECG (reais e sintéticas). A entrada do gerador é um ruído que, de acordo com os pesos da rede, terá como saída um sinal ECG sintético, que em seguida será enviado para o discriminador para realização de comparação entre a qualidade da instância gerada e a natureza de um sinal real.

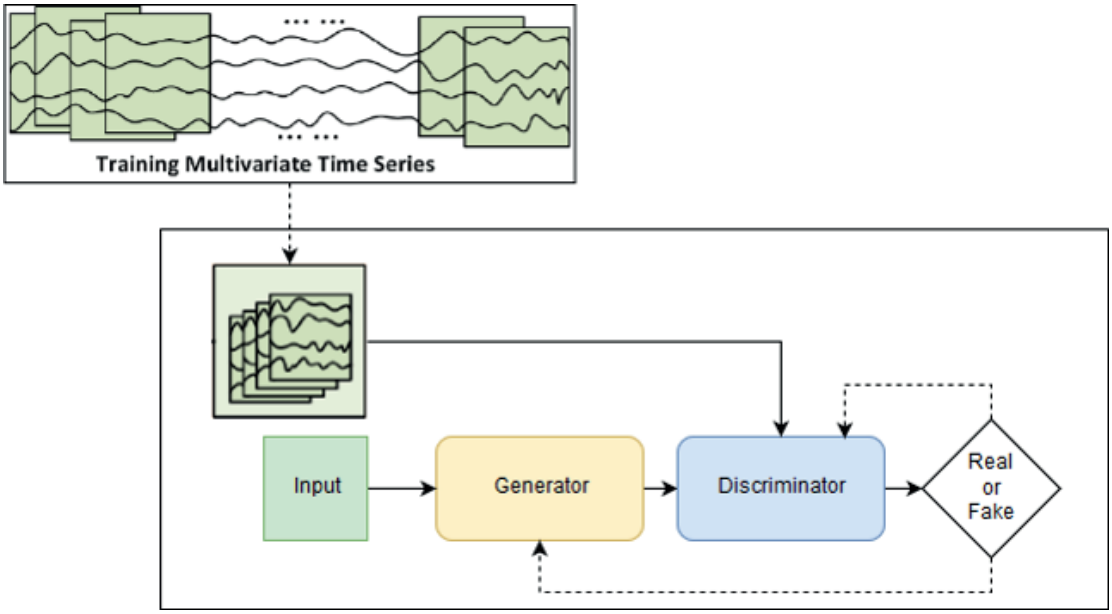


Figura 2. Rede Gerativa Adversária (GAN)

Por se tratar de um problema com série temporal, a arquitetura da GAN terá composição de Redes Neurais Recorrentes (RNNs) em seu gerador e discriminador. A rede *Long Short Term Memory* (LSTM) será a RNN avaliada inicialmente, uma vez que possui propriedades que se aplicam ao uso de série temporal. Além disso, a LSTM terá sua saída adaptada com o algoritmo SeqToSeq [Sutskever et al. 2014], visto que a saída será uma instância em série temporal, como mostra a Figura 3.

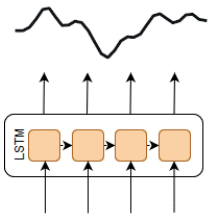


Figura 3. Rede LSTM com SeqToSeq

4.2. Dados Experimentais

Foi obtido junto ao Hospital Universitário Clementino Fraga Filho, da UFRJ, um conjunto de dados de sinais ECG reais de pacientes com doença de Chagas, incluindo-se pacientes que já faleceram de morte súbita. O conjunto de dados contém, atualmente,

400 instâncias: sinais ECG de pacientes portadores de doença de Chagas (vivos e que já faleceram) com diferentes níveis de comprometimento cardíaco. Cada sinal foi coletado durante um Holter de 24h. Na Figura 4 temos um exemplo de um pequeno trecho de sinal ECG de um paciente chagásico.

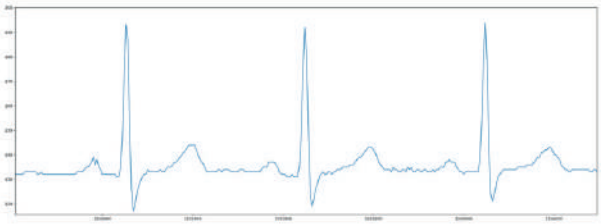


Figura 4. Exemplo de ECG do conjunto de dados

Do total de instâncias, apenas um subconjunto pequeno possui rótulos de paciente que tiveram morte súbita. Com isto, os sinais ECG destes pacientes serão divididos em intervalos menores, contendo pelo menos dois ciclos de batimento, para aumentar o número de amostras de treinamento. A escolha de dois ciclos se dá devido a garantia de ter, pelo menos, duas ondas PQRST em uma instância do conjunto de dados.

5. Conclusões

Este trabalho tem como objetivo gerar dados sintéticos de sinais ECG associados a pacientes com doença de Chagas que tiveram morte súbita, uma vez que poucos sinais reais com essas características estão disponíveis atualmente. Essa geração será realizada utilizando-se redes gerativas adversárias (GAN), contendo em sua estrutura redes LSTM.

Referências

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan. *ArXiv*, abs/1701.07875.

Bond, R. R., Finlay, D. D., Nugent, C. D., and Moore, G. (2011). A review of ecg storage formats. *International journal of medical informatics*, 80 10:681–97.

Brophy, E., Wang, Z., and E. Ward, T. (2019). Quick and easy time series generation with established image-based gans.

Celin, S. and Vasanth, K. (2017). Survey on the methods for detecting arrhythmias using heart rate signals. *Journal of Pharmaceutical Sciences and Research*, 9:183–189.

Dalvi, R. d. F., Zago, G. T., and AndreÃ, R. V. A. (2016). Heartbeat classification system based on neural networks and dimensionality reduction. *Research on Biomedical Engineering*, 32:318 – 326.

Esteban, C., Hyland, S. L., and Rätsch, G. (2018). Real-valued (medical) time series generation with recurrent conditional gans. *ArXiv*, abs/1706.02633.

Geselowitz, D. B. (1989). On the theory of the electrocardiogram. *Proceedings of the IEEE*, 77(6):857–876.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C. K., and Stanley, H. E. (2000). Physiobank,

physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101 23:E215–20.

Gonçalves, B., Guizzardi, G., and Pereira Filho, J. G. (2007). An electrocardiogram (ECG) domain ontology. In *Workshop on Ontologies and Metamodels for Software and Data Engineering, 2nd, João Pessoa, Brazil*, pages 68–81.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial nets. In *NIPS*.

Li, D. L., Chen, D., Shi, L., Jin, B., Goh, J., and Ng, S.-K. (2019). Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. *ArXiv*, abs/1901.04997.

Luz, E. J. d. S., Schwartz, W. R., Cámara-Chávez, G., and Menotti, D. (2016). Ecg-based heartbeat classification for arrhythmia detection. *Comput. Methods Prog. Biomed.*, 127(C):144–164.

Mandellos, G. J., Koukias, M. N., Styliadis, I. S., and Lymberopoulos, D. K. (2010). e-scp-ecg+ protocol: An expansion on scp-ecg protocol for health telemonitoring—pilot implementation. In *International journal of telemedicine and applications*.

Mcsharry, P., D Clifford, G., Tarassenko, L., and Smith, L. (2003). Dynamical model for generating synthetic electrocardiogram signals. *IEEE transactions on bio-medical engineering*, 50:289–94.

Nanarkar, H. M. and Chawan, P. M. (2018). A survey on classification and identification of arrythmia using machine learning techniques. *International Research Journal of Engineering and Technology*, 5:446–449.

Nash, J. (1950). Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36:48–49.

Osborne, M. J. and Rubinstein, A. (1994). *A course in game theory*. The MIT Press, Cambridge, USA. electronic edition.

Rahhal, M. M. A., Bazi, Y., Alhichri, H. S., Alajlan, N., Melgani, F., and Yager, R. R. (2016). Deep learning approach for active classification of electrocardiogram signals. *Inf. Sci.*, 345:340–354.

Ramakrishnan, S., Akshaya, V., Kishor, S., and Thyagarajan, T. (2017). Real time implementation of arrhythmia classification algorithm using statistical methods. pages 1–4.

Rassi-Junior, A., Rassi, S. G., and Rassi, A. (2000). Morte súbita na doença de chagas. *Arquivo Brasileiro de Cardiologia*, 76:75–85.

Souza, C., Andreão, R., and Segatto, M. (2006). Processamento de sinais de ecg para geração automática de alarmes.

Sutskever, I., Vinyals, O., and V. Le, Q. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 4.

Xiang, Y., Lin, Z., and Meng, J. (2018). Automatic qrs complex detection using two-level convolutional neural network. *BioMedical Engineering OnLine*, 17(1):13.

Detecção de Estresse em Sinais de EEG Utilizando Aprendizagem Profunda

Lucas Cabral[1], **José Maria Monteiro**[1], **João Alexandre Lôbo Marques**[2]

[1]Universidade Federal do Ceará (UFC)
Fortaleza, CE – Brazil

[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Shenzhen – China

lucascabral@alu.ufc.br, monteiro@dc.ufc.br, alexandre.lobo@usj.edu.mo

Nível: Mestrado
Ingresso: Fevereiro de 2019
Previsão de Término: Fevereiro de 2021
Etapas já concluídas: Revisão Bibliográfica Preliminar, Definição do Problema
Defesa da Pré-Proposta: Março de 2020
Defesa da Proposta: Dezembro de 2020

Abstract. *Prolonged maintenance of severe stress conditions causes serious physical and mental damage. This situation is aggravated in individuals in situations of high social vulnerability. A reliable method of assessing the amount of stress can provide an important tool for professionals to support people in this condition. In this research project we propose the creation of a database of EEG signs collected from people in social vulnerability, using a low-cost mobile sensor and labeled by specialists with stress levels. We also propose the development and evaluation of a deep learning method for automatic classification of this signal, in order to generate a tool of assistance to mental health professionals who provide assistance to these groups.*

Resumo. *A manutenção prolongada de estados de estresse intenso causam severos danos físicos e mentais. Essa situação é agravada em indivíduos em situação de alta vulnerabilidade social. Um método confiável de avaliação da quantidade de estresse pode fornecer uma importante ferramenta a profissionais da saúde mental no amparo a pessoas nessa condição. Neste projeto de pesquisa, propomos a criação de uma base de dados de sinais de EEG de pessoas em vulnerabilidade social, obtidos com um sensor móvel de baixo custo e rotulados com níveis de estresse por especialistas. Também propomos o desenvolvimento e avaliação de um método de aprendizagem profunda para classificação automática desse sinal, de modo a gerar uma ferramenta de áuxilio a profissionais de saúde mental que prestem assistência a estes grupos.*

1. Introdução

Estresse é uma resposta fisiológica do organismo de um indivíduo a estímulos externos adversos que desencadeiam mudanças químicas, as quais afetam os sistemas imunológico, endócrino e neurológico. Quando estressado, o organismo ativa uma resposta de "lutar ou fugir', liberando uma mistura complexa de hormônios e substâncias químicas como adrenalina, cortisol e norepinefrina para preparar o corpo para a ação física, provendo um aumento de energia e foco [Tuchweber and Bois 2007].

Em curto prazo, o estresse pode oferecer vantagens em diversas situações. Entretanto, a ativação de um estado intenso de estresse por longos períodos ou com muita frequência é extremamente danosa à saúde, causando sintomas como ritmo cardíaco e respiração acelerados, sudorese, tremores, tontura, aumento da pressão arterial, insônia, tensão muscular e dor de cabeça. Além disso, a manuntenção prolongada de estresse é considerada um fator de risco para o surgimento de condições graves como depressão, ansiedade, transtornos alimentares, hipertensão, enfraquecimento do sistema imunológico, doenças circulatórias, AVC e ataque cardíaco, além de alteração das funções cerebrais e cognitivas [Carlson 2013].

Indivíduos em situação de vulnerabilidade social e pobreza estão especialmente sujeitos a altos níveis de estresse [Silva et al. 2019]. Em grupos de risco, como vítimas de abuso, violência ou em situação de rua, essa condição se agrava. Profissionais que prestam assistência psicossocial à estes grupos possuem o desafio de avaliar a situação particular de cada indivíduo, indentificando o seu nível de estresse, sua condição psicológica e nível de risco, para então buscar a abordagem de assistência mais adequada.

A mensuração de níveis de estresse se dá principalmente por meio de questionários, cujo resultados são enviesados e subjetivos por sua própria natureza [Larson and Csikszentmihalyi 2014]. Uma análise mais objetiva pode ser feita pela mensuração dos níveis de cortisol presentes no organismo, um hormônio fortemente associado com reações de estresse. Entretanto, esses exames são caros e demorados.

O eletroencefalograma, ou EEG, é um biosinal gerado pela atividade elétrica do cérebro e captado por um conjunto de eletrodos espalhados pelo couro cabeludo. Uma vez que tem um forte impacto no estado mental, o estresse também pode ser detectado a partir do EEG [Giannakakis et al. 2015], sendo uma alternativa em fornercer uma avaliação quantitativa de níveis de estresse mais prática e rápida que exames de cortisol. Porém, um equipamento de EEG clínico possui um custo alto.

Recentemente, sensores móveis de EEG de baixo custo começaram a se popularizar. Esses equipamentos têm sido utilizados em aplicações de *neurofeedback* em educação, jogos, treino de concentração e outros. Enquanto o EEG padrão possui em torno de 20 a 40 canais, dispositivos de EEG móveis geralmente possuem bem menos. O dispositivo *Brainlink*®, por exemplo, possui apenas um canal localizado no lobo frontal, o que reduz enormemente a resolução espacial e dificulta uma extração de informações mais completas, uma vez que não contempla a atividade eletrica de todas as regiões do cortéx cerebral.

Neste contexto, este trabalho tem por objetivo propor uma arquitetura de aprendizado profundo para avaliar o stress de um indivíduo a partir do sinal eletroencefalograma captado por um sensor móvel. Dado que o lobo pré-frontal é uma região relacionada

a formação de pensamentos complexos [Davidson 2004], nossa hipótese é de que, gerando uma quantidade razoável de dados e com uma arquitetura adequada de aprendizado profundo, é possível predizer níveis de estresse a partir do sinal de EEG captado por um dispositivo de baixo custo com apenas um canal. Propomos então a coleta de dados a partir do sensor de um canal *Brainlink*®, o desenvolvimento desta arquitetura e sua avaliação. A abordagem proposta tem por finalidade auxiliar os profissionais que realizam assistência psicossocial no diagnóstico do nível de estresse em populações em situação de vulnerabilidade social.

2. Trabalhos Relacionados

Na literatura encontra-se diversas abordagens de classificação de EEG utilizando aprendizado profundo com resultados relevantes. Na tarefa de detecção de crises epilépticas, Asif et al. 2019b utiliza uma arquitetura de rede neural convolucional (CNN) chamada *SeizureNet* que aprende automaticamente os atributos e classifica o sinal de EEG inter-indivíduos. Para isso, transforma a série temporal do EEG em imagens contendo informações de tempo e frequência (espectrograma), cujos atributos serão extraidos pela rede neural. Liang et al. 2019 propõe o uso de um modelo espaço-temporal que combina CNNs com LSTMs (*Long short-term memory*), chamado LRCN, que captura a relação sequencial do sinal. Essa abordagem obteve bons resultados na base de dados aberta CHBMIT, testando em inter-pacientes com 99% de acurácia e taxa de falso-positivo de 0.2.

Uma tarefa mais relacionada a este trabalho é o reconhecimento de emoções a partir do EEG. Xu et al. 2018 faz uma revisão de sistemas de reconhecimento de emoções a partir de EEG para educação. Koelstra et al. 2011 apresenta uma base de dados multimodal para classificação de emoções a partir de estimúlos musicais, assim como resultados experimentos de classificação.

No domínio de classificação de estresse em EEG, Asif et al. 2019a compara o desempenho de diferentes classificadores, incluindo uma rede neural MLP (*Multilayer Perceptron*), na tarefa de classificar o EEG em baixo, médio e alto nível de estresse. Os sinais foram captados utilizando o dispositivo MUSE®de quatro canais. No trabalho de Jebelli et al. 2018, são utilizadas diferentes máquinas de vetores de suporte (SVM) para realizar detecção de estresse em EEG em operários de construção civil enquanto exercem atividades de risco. Os sinais são capturado a partir de um sensor móvel de baixo custo, chamado *Emotiv EPOC+*®, e são rotulados a partir de exames do nível de cortisol, sendo portanto é uma abordagem similar a proposta neste trabalho.

Não encontramos trabalhos que utilizem o dispositivo de um canal *Brainlink*®, ou focados na detecção de estresse em indivíduos de alta vulnerabilidade social.

3. A Solução Proposta

A solução proposta neste trabalho engloba a aquisição, por meio de um sensor portátil de baixo custo, e publicação de uma base de dados rotulados de sinais EEG, o desenvolvimento de uma arquitetura de aprendizado profundo para avaliar o nível de estresse de um indivíduo em situação de vulnerabilidade social, a partir do sinal EEG, e a avaliação da arquitetura proposta.

3.1. Coleta de Dados

Por meio de uma parceria já firmada com o Instituto da Primeira Infância (IPREDE) [e Italo Aguiar e Roberto Ferreira e José Carlos da Silva Filho 2017], serão coletados dados de EEG de mães em situação de alta vulnerabilidade social. O IPREDE é uma ONG cearense dedicada a promover a nutrição e o desenvolvimento na primeira infância, articulando-os com ações que visam ao fortalecimento das mulheres e da inclusão social de famílias que vivem em situação de vulnerabilidade social e pobreza.

Para a construção da base de dados, serão coletados e armazenados sinais de EEG, utilizando o dispositivo *Brainlink*®, de 30 mães voluntárias no IPREDE, que serão rotulados por especialistas da área de saúde mental. Os rótulos serão uma escala de estresse, que combina métricas obtidas através de questionários de Escala de Percepção de Stresse (EPS), uma medida global de estresse [Cohen et al. 1983], e de exames complementares de cortisol.

O dispositivo utilizado permite a coleta do sinal com uma taxa de amostragem de 512Hz. Os sinais serão coletados em ambiente controlado durante 5 minutos, onde cada voluntária estará em repouso, de olhos fechados, para reduzir ruídos. Dessa forma, serão coletadas séries temporais de 153600 amostras. Essa base de dados será disponibilizada publicamente, de modo a permitir análises por outros pesquisadores interessados.

3.2. Tratamento do Sinal

Tradicionalmente, uma etapa importante é a extração de atributos significativos do sinal EEG. Na literatura evidencia-se que não existe um único método padronizado para a extração de atributos nos sinais do EEG, que podem ser no domínio do tempo, no domínio da frequência ou uma combinação dos dois. Métodos comuns são a transformada discreta de Wavelet [Subasi 2007], análise da amplitude dos sinais [Kaper et al. 2003], métodos de agrupamento [Siuly and Wen 2010], modelagem de processo autorregressivo [Penny et al. 2000] [Pfurtscheller et al. 1998] e densidade espectral de potência [Chiappa and Bengio 2004].

Contudo, nos últimos anos, a decodificação do sinal EEG através de algoritmos de aprendizagem profunda tem recebido grande interesse da comunidade científica devido a capacidade destes em aprender representações de alto nível a partir dos dados, reduzindo a dificuldade em extrair manualmente atributos.

Para dados provenientes de sensores de baixo custo, o desafio de tratar o sinal aumenta. Em primeiro lugar, devido a sua natureza móvel, esse tipo de sensor está mais suscetível a ruídos do que o EEG clínico [Naula et al. 2017]. Em segundo lugar, o EEG clínico utiliza comumente 20 canais, enquanto que o dispositivo utilizado nesta pesquisa possui apenas um. Essa é uma perda relevante de informação, pois ignora a atividade cerebral conjunta das diferentes regiões do cortéx.

3.3. Classificação

O objetivo é realizar a classificação inter-indivíduos, ou seja, treinando o algoritmo com dados de diferentes indivíduos. Para experimentos iniciais, será implementado uma rede neural *feedforward* simples, cujo desempenho servirá de *baseline* para comparações futuras. Propomos ainda uma abordagem combinando redes neurais convolucionais com

LSTMs. O sinal puro será inicialmente pré-processado para filtragem de ruídos e transformado em imagens espectrais, contendo informação de tempo e frequência, em janelas deslizantes de tamanho a ser definido experimentalmente. Os atributos dessas imagens serão aprendidos pelas camadas de convolução da rede. Esse método possui duas vantagens importantes: em primeiro lugar, ele aumenta a quantidade de dados, uma vez que utiliza uma janela deslizante para criar imagens a partir do sinal puro; em segundo lugar, ele elimina a dificuldade de extração manual de atributos, e permite que a rede neural aprenda os atributos mais significativos a partir das imagens do sinal transformado. Após extraídos, os atributos irão alimentar a LSTM, que tem a capacidade de aprender relações sequênciais entre dados, tendo como saída a classe de nível de estresse.

A avaliação dos resultados se dará por meio de métricas tradicionais de classificação multiclasse como acurácia, precisão, *recall* e F1 *score*. Além disso, de modo a tornar o resultado mais interpretável pelo profissional que utilizará o algoritmo, propomos técnicas *Explainable Artificial Intelligence* (XAI) [Murdoch et al. 2019] através da visualização dos atributos aprendidos pela camada de convolução, identificando os atributos mais relacionados com a predição feita pela rede.

3.4. Contribuições Esperadas

Se as hipóteses formuladas mostrarem-se corretas, espera-se que este trabalho proporcione as seguintes contribuições:

1. A construção de base de dados aberta de sinais de EEG móvel de um canal, rotulados com nível de estresse. Até o momento, não temos conhecimento de nenhuma base de dados similar aberta.
2. O desenvolvimento e uma avaliação de uma arquitetura de rede neural profunda com interpretabilidade para classificação de níveis de estresse em sinal de EEG de um canal.
3. A criação de uma técnica complementar de baixo custo para auxiliar profissionais da saúde mental na assistência e tratamento de indivíduos em situação de alta vulnerabilidade social, com imediata aplicação no IPREDE.
4. A criação de uma ferramenta de pesquisa na área do estresse, fornecendo mais informações na investigação de suas causas, efeitos e tratamento.

4. Metodologia de Pesquisa

Este projeto de pesquisa consiste de diversas atividades, detalhadas a seguir:

1. Revisão bibliográfica de métodos de classificação em EEG e detecção de estresse (em andamento).
2. Desenvolvimento de software de coleta e amarzenamento de sinais de EEG, utilizando o dispositivo *Brainlink*®(em andamento).
3. Coleta de sinais e construção da base de dados aberta (não iniciado).
4. Pré-processamento dos sinais puros e transformação destes em imagens de frequência (não iniciado)
5. Criação de um *baseline* de desempenho utilizando uma rede neural feedforward (não iniciado).
6. Desenvolvimento da arquitetura proposta (não iniciado).

7. Teste e validação da arquitetura proposta e ajuste fino dos hiperparâmetros (não iniciado).
8. Desenvolvimento do algoritmo de visualização de atributos aprendidos (não iniciado).

5. Publicação dos Resultados

Espera-se a publicação de dois artigos em periódicos ou congressos em 2020, com as seguintes temáticas:

1. Descrição e análise exploratória da base de dados criada.
2. Descrição e análise dos resultados obtidos com a arquitetura de aprendizagem profunda proposta.

6. Conclusão

Esse trabalho propõe um método complementar na identificação de níveis de estresse, baseado na classificação de sinais EEG com aprendizagem profunda. Esse método pode fornecer informações valiosas na compreensão e tratamento de estresse, contribuindo para uma melhora na qualidade de vida de indivíduos e famílias atendidos por profissionais de saúde mental. Além disso, o método possui um custo relativamente baixo e grande mobilidade, permitindo que profissionais o apliquem em campo, atendendo em localidades rurais e isoladas. Assim, o método proposto possui relevância científica e social.

Referências

[Asif et al. 2019a] Asif, A., Majid, M., and Anwar, S. M. (2019a). Human stress classification using eeg signals in response to music tracks. *Computers in Biology and Medicine*, 107:182 – 196.

[Asif et al. 2019b] Asif, U., Roy, S., Tang, J., and Harrer, S. (2019b). Seizurenet: A deep convolutional neural network for accurate seizure type classification and seizure detection.

[Carlson 2013] Carlson, N. (2013). *Physiology of Behavior*. Always learning. Pearson.

[Chiappa and Bengio 2004] Chiappa, S. and Bengio, S. (2004). HMM and IOHMM modeling of EEG rhythms for asynchronous BCI systems. *European Symposium on Artificial Neural Networks ESANN*.

[Cohen et al. 1983] Cohen, S., Kamarck, T., and Mermelstein, R. (1983). A global measure of perceived stress. *Journal of Health and Social Behavior*, 24(4):385–396.

[Davidson 2004] Davidson, R. J. (2004). What does the prefrontal cortex "do" in affect: perspectives on frontal eeg asymmetry research. *Biological Psychology*, 67(1):219 – 234. Frontal EEG Asymmetry, Emotion, and Psychopathology.

[e Italo Aguiar e Roberto Ferreira e José Carlos da Silva Filho 2017] e Italo Aguiar e Roberto Ferreira e José Carlos da Silva Filho, B. L. (2017). The benefits of cooperation between university, ngos and communities – the case of iprede in ceará. *Revista de Ciências da Administração*, 19(49):74–85.

[Giannakakis et al. 2015] Giannakakis, G., Grigoriadis, D., and Tsiknakis, M. (2015). Detection of stress/anxiety state from eeg features during video watching. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6034–6037.

[Jebelli et al. 2018] Jebelli, H., Hwang, S., and Lee, S. (2018). Eeg-based workers' stress recognition at construction sites. *Automation in Construction*, 93:315 – 324.

[Kaper et al. 2003] Kaper, M., Meinicke, P., Grossekathoefer, U., Lingner, T., and Ritter, H. (2003). BCI competition 2003-data set iib: support vector machines for the p300 speller paradigm. *IEEE Trans. Biomed. Eng.*, 51. pp. 1073-1076.

[Koelstra et al. 2011] Koelstra, S., Mühl, C., Soleymani, M., Lee, J.-S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., and Yiannis) Patras, I. (2011). Deap: A database for emotion analysis using physiological signals. *IEEE Transactions on Affective Computing*, 3:18–31.

[Larson and Csikszentmihalyi 2014] Larson, R. and Csikszentmihalyi, M. (2014). *The Experience Sampling Method*, pages 21–34. Springer Netherlands, Dordrecht.

[Liang et al. 2019] Liang, W., Pei, H., Cai, Q., and Wang, Y. (2019). Scalp eeg epileptogenic zone recognition and localization based on long-term recurrent convolutional network. *Neurocomputing*.

[Murdoch et al. 2019] Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Interpretable machine learning: definitions, methods, and applications. *arXiv e-prints*, page arXiv:1901.04592.

[Naula et al. 2017] Naula, E., García, A. F., Palacio-Baus, K., Minchala, L. I., Vazquez-Rodas, A., and Astudillo, D. (2017). Evaluating the mindwave headset for automatic upper body motion classification. In *2017 International Conference on Information Systems and Computer Science (INCISCOS)*, pages 166–173.

[Penny et al. 2000] Penny, W. D., Roberts, S. J., Curran, E., and Stokes, M. J. (2000). EEG-based communication: a pattern recognition approach. *IEEE Trans. Rehabil. Eng*, 8. pp. 214-215.

[Pfurtscheller et al. 1998] Pfurtscheller, G., Neuper, C., Schlogl, A., and Lugger, K. (1998). Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. *IEEE Trans. Rehabil. Eng*, 6. pp. 316-355.

[Silva et al. 2019] Silva, d. C. P. d., Cunha, K. d. C., Ramos, E. M. L. S., Pontes, F. A. R., and Silva, S. S. d. C. (2019). Estresse parental em famílias pobres. *Psicologia em Estudo*, 24.

[Siuly and Wen 2010] Siuly, Y. L. and Wen, P. (2010). Clustering technique-based least square support vector machine for EEG signal classification. *Comput. Methods Programs Biomed.*

[Subasi 2007] Subasi, A. (2007). EEG signal classification using wavelet feature extraction and a mixture of expert mode. *Expert Systems with Applications*, 32. pp. 1084-1093.

[Tuchweber and Bois 2007] Tuchweber, B. and Bois, P. (2007). Selye, hans*. In Fink, G., editor, *Encyclopedia of Stress (Second Edition)*, pages 448 – 450. Academic Press, New York, second edition edition.

[Xu et al. 2018] Xu, T., Zhou, Y., Wang, Z., and Peng, Y. (2018). Learning emotions eeg-based recognition and brain activity: A survey study on bci for intelligent tutoring system. *Procedia Computer Science*, 130:376 – 382. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.

Um Ambiente de Desenvolvimento de Sistemas de Armazenamento para Sensores

Alexandre R. Ordakowski[1]
Orientadora: Carmem S. Hara[1]
Coorientador: Marcos A. Carrero[1]

[1]DINF – Universidade Federal do Paraná – UFPR – Paraná, Brasil
{arordakowski,carmem,macarrero}@inf.ufpr.br

Nível: Mestrado
Ingresso no Programa: Agosto/2018
Previsão de Conclusão: Agosto/2020
Etapas Concluídas: Créditos em disciplinas, Implementação inicial
Etapas Futuras: Exame de qualificação, Defesa da dissertação

Resumo. *Nos últimos anos, o número de dispositivos de sensoriamento em ambientes urbanos aumentou significativamente. O crescente volume de dados gerados exige que novos modelos de armazenamento sejam desenvolvidos para as Redes de Sensores Sem Fio (RSSF). A especificação e implementação desses modelos de sistema é complexa e poucos trabalhos focam em ferramentas que dão apoio ao seu desenvolvimento. Assim, nesta dissertação é proposta a construção de um ambiente que forneça suporte ao desenvolvimento de sistemas de armazenamento para RSSF. Para isso, são propostos o RCBM-S, um arcabouço de componentes de software reutilizáveis para aplicações em RSSF, e a geração do código para dispositivos sensores a partir do SLEDS, uma linguagem de domínio específico (DSL, do inglês Domain Specific Language) voltada para a coordenação entre os componentes.*

Palavras-chave: *Linguagem de domínio específico, sistemas de armazenamento de dados, redes de sensores sem fio.*

1. Introdução

O uso de tecnologias para minimizar os problemas encontrados nos grandes centros urbanos e promover o desenvolvimento sustentável deu origem ao termo "Cidades Inteligentes" [Yin et al. 2015]. As redes de sensores sem fio (RSSF) são uma das principais tecnologias utilizadas para esse fim, servindo como fonte de coleta e disseminação de dados. Conforme o número de dispositivos de sensoriamento de ambientes urbanos cresce, bem como o número de aplicações, novos sistemas de armazenamento de dados são exigidos. Porém, o desenvolvimento de tais sistemas não é uma tarefa simples, visto que, devido aos recursos limitados dos dispositivos cada sistema de armazenamento e disseminação de dados deve ser planejado para se adequar à um uso específico.

Os modelos de armazenamento de dados em sensores podem ser classificados como de armazenamento **externo** (centralizado) ou **na rede** (distribuído). No armazenamento externo, os dados de sensoriamento são enviados a uma base de dados externa, ou estação-base (EB). Nas RSSFs de grande escala, manter todo o processamento centralizado aumenta o custo de comunicação [Can and Demirbas 2013], sendo menos escalável que abordagens na rede [Coman et al. 2007]. Assim, os dados coletados podem ser agrupados em *repositórios* de dados distribuídos na rede. Uma técnica comum é a formação de agrupamentos (*clusters*) de sensores e a eleição de um ou mais sensores como líderes (*cluster-heads* - CH) do grupo [Amaxilatis et al. 2011], que armazenam os dados dos membros do agrupamento (*cluster members*), e respondem as requisições de consultas. Este artigo concentra-se em modelos de armazenamento na rede devido a sua escalabilidade.

Poucos trabalhos na literatura apresentam abordagens que tratem da modelagem e implementação de sistemas de armazenamento de dados em RSSF de *maneira sistemática*, requisito essencial para apoiar o desenvolvimento de tais sistemas. Observando essa lacuna, [Carrero et al. 2017] consideram ambientes de *simulação* de RSSF e propõem o *framework* RCBM (*Reusable Component-Based Model*) e a linguagem de domínio específico (DSL) SLEDS (*State Machine-based Language for Event-Driven Systems*) [Carrero et al. 2018]. O RCBM baseia-se na definição de componentes reutilizáveis para geração de *códigos de simulação* de sistemas de armazenamento de dados em RSSFs. A linguagem SLEDS é baseada em máquinas de estados e especifica a orquestração dos componentes. O objetivo desta dissertação é adotar as mesmas estratégias de desenvolvimento para a geração de código para dispositivos sensores em *ambientes de sensoriamento reais*. O objetivo é desenvolver o RCBM-S (RCBM - *for Sensor devices*), e a tradução da linguagem SLEDS para nesC, que é uma das linguagens mais utilizadas para o desenvolvimento de sistemas para dispositivos sensores. Assim, o RCBM-S e o SLEDS compõem um ambiente completo de desenvolvimento de sistemas de armazenamento de dados em *ambientes de sensoriamento reais*.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta trabalhos relacionados. O *framework* RCBM-S e a linguagem SLEDS são apresentados na Seção 3. A Seção 4 apresenta o estado atual do trabalho e a Seção 5 finaliza o artigo.

2. Trabalhos Relacionados

Propostas existentes na literatura que possuem objetivos semelhantes ao RCBM-S geram código para sistemas operacionais específicos de RSSF, como o TinyOS e Contiki, ou usando uma biblioteca independente de plataforma, como o WiseLib [Baumgartner et al. 2010], que através da utilização de interfaces genéricas consegue gerar código para plataformas distintas.

Trabalhos que aplicam separação de conceitos, como SenNet [Salman and Al-Yasiri 2016], MDDWSN [Tei et al. 2014] e IoTSuite [Patel and Cassou 2015], destacam aspectos específicos do sistema em construção. O MDDWSN e o SenNet propõem um processo de desenvolvimento de software para abordar conceitos relacionados ao processamento de dados e relacionados à rede. Essas ferramentas, no entanto, não suportam o conceito de repositórios de dados distribuídos. IoTSuite, por outro lado, especifica os conceitos comuns presentes nos cenários de IoT, dando liberdade para o desenvolvedor adaptar a gramática da linguagem utilizada no *framework* de acordo com o domínio específico. O *Communicating X-Machine* (CXM) [de Lima Braga et al. 2010], propõe um método baseado em máquinas de estados para especificar o comportamento reativo de sistemas em RSSF. Entretanto, as transições de estados propostas nesta ferramenta são realizadas apenas por eventos, sem propor uma transição resultante de um processamento lógico no estado.

O RCBM (*Reusable Component-Based Model*) [Carrero et al. 2017] é um *framework* que propõe o uso de componentes de software reutilizáveis para desenvolvimento sistemas de armazenamento de dados em repositórios. Complementando o RCBM, a DSL SLEDS especificada por [Carrero et al. 2018], é uma linguagem de alto nível que abstrai detalhes específicos da programação da orquestração de componentes na *plataforma de simulação* NS-2. Embora a simulação seja uma ferramenta importante para validação de sistemas, o objetivo final é sempre a implantação da aplicação em dispositivos sensores e em cenários reais. Assim, esta dissertação tem como objetivo investigar a adoção de uma abordagem semelhante à adotada pelo *framework* RCBM e a linguagem SLEDS para dar suporte ao desenvolvimento de modelos de armazenamento para *sensores em ambientes de sensoriamento reais*.

3. Solução Proposta

Visando fornecer suporte para o desenvolvimento de modelos de armazenamento de dados em RSSF, a pesquisa almeja atingir duas metas. A primeira é o desenvolvimento do *framework* RCBM-S. O RCBM-S trata-se de um modelo baseado em componentes reutilizáveis para desenvolvimento de aplicações de RSSF em *ambientes reais*. A segunda meta é a geração de código para sensores a partir da DSL SLEDS, visando auxiliar a especificação e programação do coordenador dos componentes para plataformas reais.

3.1. RCBM-S

O *framework* RCBM-S [Ordakowski et al. 2019] foi desenvolvido para auxiliar o desenvolvedor a reutilizar componentes de software por meio da modularização de funcionalidades de determinadas entidades. O uso do RCBM-S permite que o código específico dos sistemas de armazenamento de dados em RSSF sejam separados do código que define o fluxo de execução. O RCBM-S utiliza três tipos distintos de componentes reutilizáveis, separados por suas finalidades específicas.

O **Componente de aplicação** é definido por funcionalidades comuns de entidades das aplicações. Em sistemas de armazenamento de dados baseados em *repositórios*, a rede realiza a escolha de *Cluster-Heads* (CH), que são sensores responsáveis por centralizar as leituras de um grupo de sensores. Os grupos, são chamados de *clusters*. Dessa forma, as funcionalidades básicas da entidade ***Cluster*** são a eleição dos CHs e a associação de sensores membros ao grupo liderado por um CH. Na Listagem 1 essas funcionalidades são representadas pelos comandos *selectCH* e *join* respectivamente.

```
1 interface Cluster {
2   command int selectCH( int neighbors[] );
3   command int join( int candidates[] );}
```

Listing 1. Template do componente Cluster.

Outra modalidade de componente do RCBM-S são os **componentes de biblioteca**. Este tipo de componente não está diretamente relacionado a nenhuma entidade, mas são genéricos e auxiliam na codificação de aplicações em geral. Exemplificando, a Listagem 2 apresenta a interface do componente biblioteca de agregação.

```
1 interface Library_Aggregation {
2     command int max( double val[]  ] );
3     command int min( double val[] );
4     command double avg( double val[ ] );
5     command int sum( int val[] ); }
```

Listing 2. Interface do componente Library_Aggregation.

Como a implementação do fluxo de execução da aplicação é um código específico, o desenvolvedor raramente se preocupa com a reutilização do código por outros sistemas. Entretanto, ao realizar a especificação utilizando um método formal, o programador pode facilmente reaproveitar o código para novas implementações. O orquestrador do fluxo de execução do RCBM-S é o **componente do coordenador**. A linguagem SLEDS [Carrero et al. 2018] foi desenvolvida para definir a coordenação de componentes utilizando máquinas de estados, visando a melhor representação do comportamento lógico e reativo das aplicações em RSSFs. Um dos objetivos desta dissertação é gerar o código em nesC para a coordenação de componentes do RCBM-S a partir de um programa na linguagem SLEDS.

3.2. A Linguagem SLEDS

A linguagem SLEDS [Carrero et al. 2018] foi desenvolvida utilizando como método formal as máquinas de estados. Este modelo é adequado para especificar eventos ou interações entre os componentes de aplicação do RCBM-S. A Figura 1 é a representação da máquina de estados do fluxo de execução de um sistema de descoberta de vizinhos por inundação. A aplicação inicia pelo estado **INI**, que faz uma busca pelo sensor com o identificador único (ID) com valor 0. Após encontrá-lo, o sensor com ID 0 envia o seu id em um *broadcast* e realiza a transição de estados lógica para o estado **Wait_First_Sensor_ID**. Quando o sensor recebe uma mensagem, o mesmo envia o seu ID e faz uma transição de estados por evento para o estado **Form_Neighbor_List**. Neste estado, a aplicação fica recebendo os IDs dos seus vizinhos por um determinado tempo. Em seguida, ocorre outra transição por evento para o estado **ACK_Neighbor_List**. Neste estado, os sen-

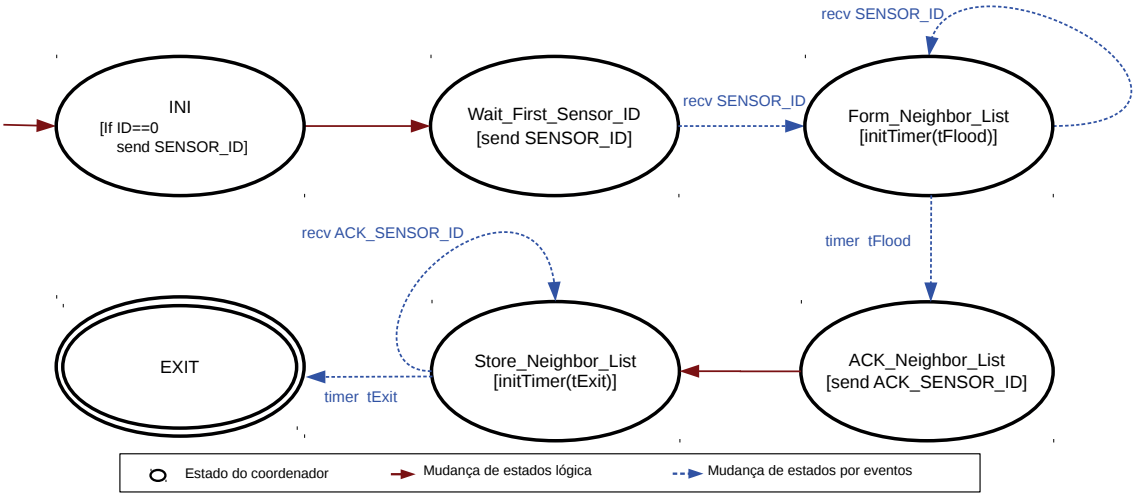


Figura 1. Um modelo de máquina de estados para descoberta de vizinhos por inundação. [Carrero et al. 2017].

sores enviam uma mensagem de confirmação para seus vizinhos reconhecidos e realizam a transição lógica para o estado **Store_Neighbor_List**, onde recebem mensagens de ACK_SENSOR_ID dos seus vizinhos e armazenam seus IDs. Por fim, após um tempo recebendo mensagens de confirmação, a aplicação realiza a transição para o estado **EXIT**.

Um programa SLEDS é composto pela descrição de um conjunto de estados, que por sua vez, consiste em uma sequência de ações. Um exemplo pode ser encontrado na Listagem 3. Este código SLEDS representa o estado *Form_Neighbor_List*. O comando *during() on recv()* (linha 2) é a expressão em SLEDS que indica a inicialização de um *timer* e o recebimento de mensagens. Neste estado, durante um tempo *tFlood* a aplicação fica recebendo mensagens. A cada novo recebimento, o Id recebido na mensagem é armazenado em uma lista (linha 3). Por fim, quando o *timer* expira, é realizada uma transição para o estado *ACK_Neighbor_List()*.

```
1 STATE Form_Neighbor_List() {
2  during (tFlood) on recvBroadcast(SENSORID, msdID, Id){
3   listSensorAnnouncements.insert(Id); }
4  nextState ACK_Neighbor_List()}
```

Listing 3. Implementação do estado Form_Neighbor_List em SLEDS.

O SLEDS foi proposto para a geração de códigos de *simulação* para aplicações de armazenamento de dados em RSSF. Assim, nesta dissertação, uma questão a ser investigada é se ela também é adequada para a especificação e geração de código para *dispositivos sensores*. Uma possível tradução do código em SLEDS na Listagem 3 para nesC é apresentada na Listagem 4. O *STATE Form_Neighbor_List()* em SLEDS foi traduzido para o método *state_Form_Neighbor_List()*. Dessa maneira, a transição de estados lógica do SLEDS pode ser implementada por uma chamada de método em nesC.

```
1 void state_Form_Neighbor_List(){
2     currentState = FORM_NEIGHBOR_LIST;
3     call Timer.startOneShot(tFlood) ; }
4
5 event message_t* Receive.receive(message_t* msg, void* payload){
```

```
6       SENSOR_ID* btrpkt = (SENSOR_ID*)payload;
7       if(currentState == FORM_NEIGHBOR_LIST){
8           listSensorAnnouncements.insert(btrpkt->Id); }}
9
10  event void Timer.fired(){
11      if(currentState == FORM_NEIGHBOR_LIST){
12          state_ACK_Neighbor_List(); }}
```

Listing 4. Implementação do estado Form_Neighbor_List em nesC.

As transições por eventos acontecem de maneira reativa. Ou seja, são uma reação à ocorrência de uma determinada ação. A Listagem 4 apresenta dois eventos que provocam transições: o *Timer.fired*, acionado ao final de um *timer* e o *Receive.receive*, acionado no recebimento de uma mensagem. Para realizar a transição em ambos os casos, o primeiro passo é verificar qual o estado atual (linhas 7 e 11), para assim decidir as ações a serem tomadas.

4. Estado atual e atividades

Atualmente, o *framework* RCBM-S para a geração de código na linguagem nesC foi concluído. Os resultados da utilização do RCBM-S no desenvolvimento de duas aplicações de repositório de dados apontam que cerca de 80% das linhas de códigos das aplicações foram reutilizadas, o que indica o potencial da proposta [Ordakowski et al. 2019]. As fases do desenvolvimento da pesquisa que já foram concluídas são: levantamento bibliográfico; elaboração da proposta; desenvolvimento do *framework* RCBM-S; análise léxica e sintática da linguagem SLEDS. A Tabela 1 apresenta o cronograma das atividades futuras.

	2019			2020			
Atividade	08	09-10	11-12	1-2	3-4	5-6	7-8
Análise semântica do SLEDS	X	X	X				
Exame de Qualificação		X					
Geração de código nesC		X	X	X	X		
Validação com experimentos			X	X	X		
Escrita de artigos					X	X	X
Escrita da dissertação			X	X	X	X	X
Defesa da dissertação							X

Tabela 1. Cronograma de atividades

A análise semântica do SLEDS será realizada através das ferramentas Flex e Bison, por meio da construção da árvore de *tokens*. Em seguida, após o exame de qualificação da proposta, será realizada a geração de códigos em nesC utilizando a árvore semântica. A validação da proposta envolverá: (i) avaliação do números de linhas reutilizadas em aplicações de armazenamento em RSSF, e (ii) avaliação da qualidade do código gerado. Os resultados da validação serão descritos na dissertação e em artigos que devem ser submetidos futuramente.

5. Considerações Finais

Esta pesquisa apresentou o **RCBM-S**, um arcabouço baseado em componentes reutilizáveis desenvolvido para dar suporte ao desenvolvimento de sistemas de armazenamento

de dados em RSSF. Este modelo permite a separação do código específico, facilitando assim o reuso do código comum presente em grande parte dos sistemas de armazenamento de dados. O SLEDS é uma linguagem de alto nível baseada em máquinas de estados para a geração do código do componente de orquestração do RCBM-S. Como trabalho futuro é esperado que a construção do compilador do SLEDS, em conjunto com o modelo de componentes do RCBM-S, formem um ambiente completo de desenvolvimento de sistemas de armazenamento de dados em RSSF.

Referências

Amaxilatis, D., Chatzigiannakis, I., Koninis, C., and Pyrgelis, A. (2011). Component based clustering in wireless sensor networks. *arXiv preprint arXiv:1105.3864*.

Baumgartner, T., Chatzigiannakis, I., Fekete, S., Koninis, C., Kroller, A., and Pyrgelis, A. (2010). Wiselib: A generic algorithm library for heterogeneous sensor networks. In *European Conference on Wireless Sensor Networks*, pages 162–177. Springer.

Can, Z. and Demirbas, M. (2013). A survey on in-network querying and tracking services for wireless sensor networks. *Ad Hoc Networks*, 11(1):596–610.

Carrero, M. A., Musicante, M. A., dos Santos, A. L., and Hara, C. S. (2017). A reusable component-based model for wsn storage simulation. In *Proceedings of the 13th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 31–38.

Carrero, M. A., Musicante, M. A., dos Santos, A. L., and Hara, C. S. (2018). Sleds: A dsl for data-centric storage on wireless sensor networks. In *Workshop on Big Social Data and Urban Computing*, pages 74–89. Springer.

Coman, A., Sander, J., and Nascimento, M. A. (2007). Adaptive processing of historical spatial range queries in peer-to-peer sensor networks. *Distributed and Parallel Databases*, 22(2-3):133–163.

de Lima Braga, M., de Jesus dos Santos, A., and de Lucena Junior, V. F. (2010). Modelagem e geração de código para redes de sensores sem fio usando communicating x-machine. In *Proc. of the 9th Int. Information and Telecommunication Technologies Symposium*.

Ordakowski, A., Carrero, M. A., Musicante, M., dos Santos, A., and Hara, C. (2019). Desenvolvimento de modelos de armazenamento em sensores com reutilização de código. In *SBBD 2019 - Short and Vision and Industrial Papers*.

Patel, P. and Cassou, D. (2015). Enabling high-level application development for the internet of things. *Journal of Systems and Software*, 103:62–84.

Salman, A. J. and Al-Yasiri, A. (2016). Sennet: a programming toolkit to develop wireless sensor network applications. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–7.

Tei, K., Shimizu, R., Fukazawa, Y., and Honiden, S. (2014). Model-driven-development-based stepwise software development process for wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(4):675–687.

Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., and David, B. (2015). A literature survey on smart cities. *Science China Information Sciences*, 58(10):1–18.

MIRP: Uma abordagem inteligente para gerenciamento de *buffer* em banco de dados

Gustavo Moraes[1], Angelo Brayner[1], José de Aguiar Moraes Filho[2]

[1]Universidade Federal do Ceará
Fortaleza – CE – Brasil

gustavo.moraes@alu.ufc.br, brayner@dc.ufc.br

[2]Universidade de Fortaleza
Fortaleza – CE – Brasil

jaguiar@unifor.br

Abstract. *In spite of technological advances, database managers are not always able to keep all data in main memory. The buffer manager is the software component responsible for using strategies that perform data replacement between main memory and secondary storage media when necessary. Replacement strategies should consider the workload generated by user and application queries and thereby improve performance. However, strategies proposed in the literature do not always cope with workload variations very well. This paper proposes the MIRP (Most Important Replacement Policy) algorithm in order to find out workload access patterns using machine learning techniques.*

Informações Gerais

- **Título:** MIRP: Uma abordagem inteligente para gerenciamento de *buffer* em banco de dados
- **Nível:** Mestrado
- **Nome do estudante:** Gustavo de Oliveira Moraes - gustavo.moraes@alu.ufc.br
- **Orientador:** Angelo Roncalli Alencar Brayner - brayner@dc.ufc.br
- **Coorientador:** José de Aguiar Moraes Filho - jaguiar@unifor.br
- **Universidade / Programa:** Universidade Federal do Ceará / Programa de Mestrado e Doutorado em Ciência da Computação
- **Data de ingresso / Data prevista da Defesa:** Março 2019 / Agosto 2021
- **Etapas concluídas:** Revisão bibliográfica

1. Introdução

O gerenciador de *buffer* (descrito na Seção 2.1) é o componente de software de um SBBD responsável por alocar e desalocar a área de memória principal usada no momento do processamento dos dados e, consequentemente, reduzir o acesso à mídia secundária. Contudo, nem sempre é possível manter todos os dados em memória. Cabe ao gerenciador de *buffer* estabelecer estratégias que otimizem a substituição dos dados da memória principal na medida que os dados são solicitados [Effelsberg and Haerder 1984].

A Figura 1 mostra o número de instruções e ciclos de CPU realizadas pelos principais componentes de um SGBD ao executar a transação *New Order* do *benchmark* TPC-C [TPC 2010]. Os quadros brancos dos gráficos de instruções e ciclos (com valor de 6,8% e 12,3%, respectivamente) representam o trabalho real da execução da consulta. O gerenciador de *buffer* possui um impacto maior do que os demais componentes devido a criação, busca e acesso de dados. Realizar otimizações nesse componente pode trazer ganhos de desempenho ao SGBD [Harizopoulos et al. 2008].

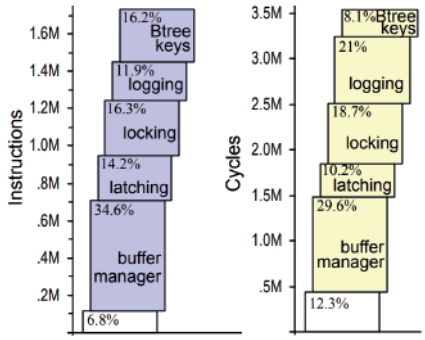


Figura 1. Impacto de uma consulta OLTP aos componentes de um SGBD [Harizopoulos et al. 2008]

Este trabalho busca elaborar estratégias de substituição eficientes utilizando aprendizagem de máquina para realizar predições das solicitações futuras e adaptar a estratégia. As próximas Seções estão estruturados da forma a seguir. A Seção 2 apresenta um referencial teórico para o entendimento do problema. A Seção 3 apresenta o problema. A Seção 4 apresenta alguns dos principais trabalhos relacionados. A Seção 5 propõe uma possível solução. A Seção 6 descreve a metodologia a ser utilizada. A Seção 7 expõe possíveis trabalhos científicos a serem publicados. E por último, a Seção 8 apresenta conclusões, e perspectivas de continuação deste trabalho.

2. Referencial Teórico

2.1. Gerenciamento de *Buffer*

O *buffer pool* é um espaço de memória que é solicitado pelo gerenciador de *buffer* ao sistema operacional. O *buffer pool* pode ser implementado por uma matriz de *frames* (região física da memória para alocação de blocos). Além disso, o gerenciador de *buffer* utiliza tabelas *hash* para o mapeamento de páginas. Cada página possui um ponteiro para seu respectivo *frame* e alguns metadados, como *dirty bit*, que indica que a página foi escrita ou sofreu apenas leitura [Hellerstein et al. 2007].

O gerenciador de *buffer* trabalha principalmente com os componentes de processamento de consulta e gerenciadores de arquivos. A Figura 2 descreve alguns processos do seu funcionamento. Quando uma consulta (por exemplo, na linguagem SQl) é submetida, o processador de consulta então realiza suas etapas de processamento; E ao executá-la, solicita uma serie de requisições de páginas ao gerenciador de *buffer*. Caso as páginas não se encontrem no *buffer pool*, o gerenciador de *buffer* requisita blocos ao gerenciador de arquivos que então finalmente realiza o acesso à mídia de armazenamento física [Hellerstein et al. 2007].

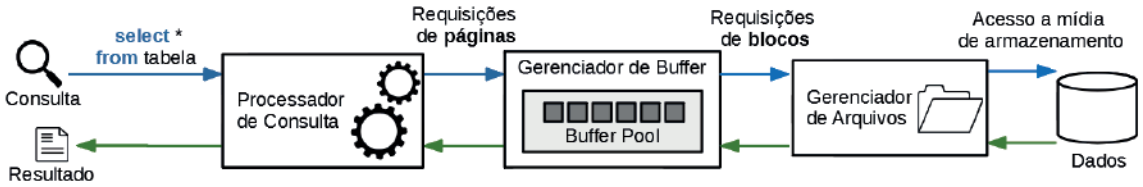


Figura 2. Principais interações do gerenciador de *buffer*

Todavia, se o *buffer pool* se encontra cheio, sem espaço suficiente para a alocação da página solicitada, o gerenciador de *buffer* utiliza uma política de substituição de páginas, que deve remover do *buffer* a página com menor valor. Várias políticas já foram apresentadas como: LRU (*Least-Recently Used*), MRU (*Most-Recently Used*), LFU (*Least-Frequently Used*), FIFO (*First in First out*) cada uma com diferentes estratégias e custos [Silberschatz et al. 2005].

2.2. Meta-aprendizagem

A Meta-aprendizagem busca encontrar dinamicamente a melhor estratégia que pode melhorar a aprendizagem dos algoritmos ao induzir caminhos alternativos na sua execução. Cada algoritmo de aprendizagem é baseado em um conjunto de suposições sobre os dados, isso significa que o resultado verificado após um processo de aprendizagem é diretamente associado com a sua maneira de resolução, caso o tipo do problema ou os dados mudem, deve ser necessário um novo processo de aprendizagem para resolvê-lo. Meta-aprendizagem é muito utilizada para automatizar a escolha de valores de parâmetros, e também avaliar o impacto de cada escolha. [Vilalta and Drissi 2002].

WMA (*Weighted Majority Algorithm*) [Littlestone and Warmuth 1994] é uma técnica de meta-aprendizagem que consiste em resolver um problema com um conjunto de algoritmos em tempo real que produzem erros e acertos ao analisar um evento. Cada algoritmo pode ser importante em diferentes momentos, um algoritmo mestre é usado para determinar qual o algoritmo ideal na resolução do evento. Existem várias variações do WMA, que podem trabalhar com alvos móveis, conjuntos infinitos ou previsões aleatórias.

3. Definição do Problema

Na execução de consultas é comum a geração de diferentes planos de execução que podem gerar diferentes padrões de acesso aos dados. Dessa forma, a carga de trabalho de um SGBD é dinâmica, pois é dependente do usuário e aplicações. Os padrões de acesso impactam diretamente no desempenho do gerenciador de *buffer*. Por exemplo, um padrão de leitura sequencial pode ocupar toda a memória rapidamente, removendo dados frequentes que são importantes durante o processamento. Elaborar es-

tratégias de substituição adaptativas aos padrões de acesso são essenciais para manter um bom desempenho ao reduzirem o acesso à mídia de armazenamento secundária [Kabra and DeWitt 1998, Megiddo and Modha 2003].

4. Trabalhos Relacionados

ACME (*Adaptive Caching using Multiple Experts*) [Ari et al. 2002] é uma estratégia de substituição de propósito geral (porém com boas aplicações em serviços de cache da Web) que utiliza várias políticas de substituição executadas simultaneamente, cada política possui um peso, que é usado para indicar qual a melhor política naquele determinado instante de tempo. Cada uma das políticas recomenda quais páginas/blocos devem ser mantidas ou removidas da memória, podendo atribuir valores para as páginas/blocos mais importantes. Por exemplo, o algoritmo LFU mantém um contador de referências para cada acesso feito na página/bloco. Caso necessite de uma substituição, a página/bloco com a menor soma de votos ponderados pelo peso da sua política respectivamente é escolhida como vítima. Os pesos de cada política são ajustados usando os algoritmos de aprendizagem de máquina como o *Weighted Majority* [Littlestone and Warmuth 1994].

ACME-DB (*Adaptive Caching using Multiple Experts for Database buffers*) [Riaz-ud Din and Kirchberg 2006] procura adaptar o ACME para o uso em em sistemas de banco de dados, realizando modificações, como tornar fixo o tamanho de cada *frame* e o uso de políticas desenvolvidas para banco de dados.

5. Solução Proposta

Neste trabalho é proposto uma política de substituição de páginas para banco de dados denominado de MIRP *(Most Important Replacement Policy)* que é baseado na estratégia ACME. Diferente do ACME, ao identificar padrões de acesso, MIRP procura adaptar o seu comportamento em resposta do padrão de acesso futuro. A Figura 3 apresenta um resumo da estrutura do MIRP. A estratégia pode ser descrita em três componentes principais:

1. *Buffer pool*: Consiste na estrutura de dados responsável pelo armazenamento físico dos dados na memória.
2. *Policy pool*: Um conjunto de Políticas Virtuais (PV) de substituição ou componentes delas, cada elemento do conjunto pode ser escolhido para a atender como uma resposta de um determinado padrão de acesso ou a combinação simplificada deles.
3. Mecanismo de Predições: Diferente do ACME, o MIRP utiliza um grupo de algoritmos e estruturas de dados que tem como objetivo principal identificar padrões e antecipar o funcionamento da *policy pool*. O mecanismo de predições é executado de forma paralela ao *buffer pool* e *policy pool*.

Inicialmente o processador de consulta realiza requisições ao *buffer pool*, as páginas são buscadas. Se uma página solicitada já se encontra no *buffer pool* (*hit* ou acerto), então é verificado por meio de ponteiros se essa página está sendo referenciada logicamente por alguma das PV mantidas pela *policy pool*. Caso esteja, o comportamento de um *hit* na página referenciada é chamado para cada PV que mantém logicamente a página. Na Figura 3 podemos observar que cada página na *buffer pool*, contém N ponteiros, que representam as N possibilidades de cada PV. As páginas ilustradas com cor

branca são páginas que já foram despejadas para a mídia de armazenamento secundária e que permanecem de forma lógica nas políticas (São preservadas apenas suas referências sem os seus dados).

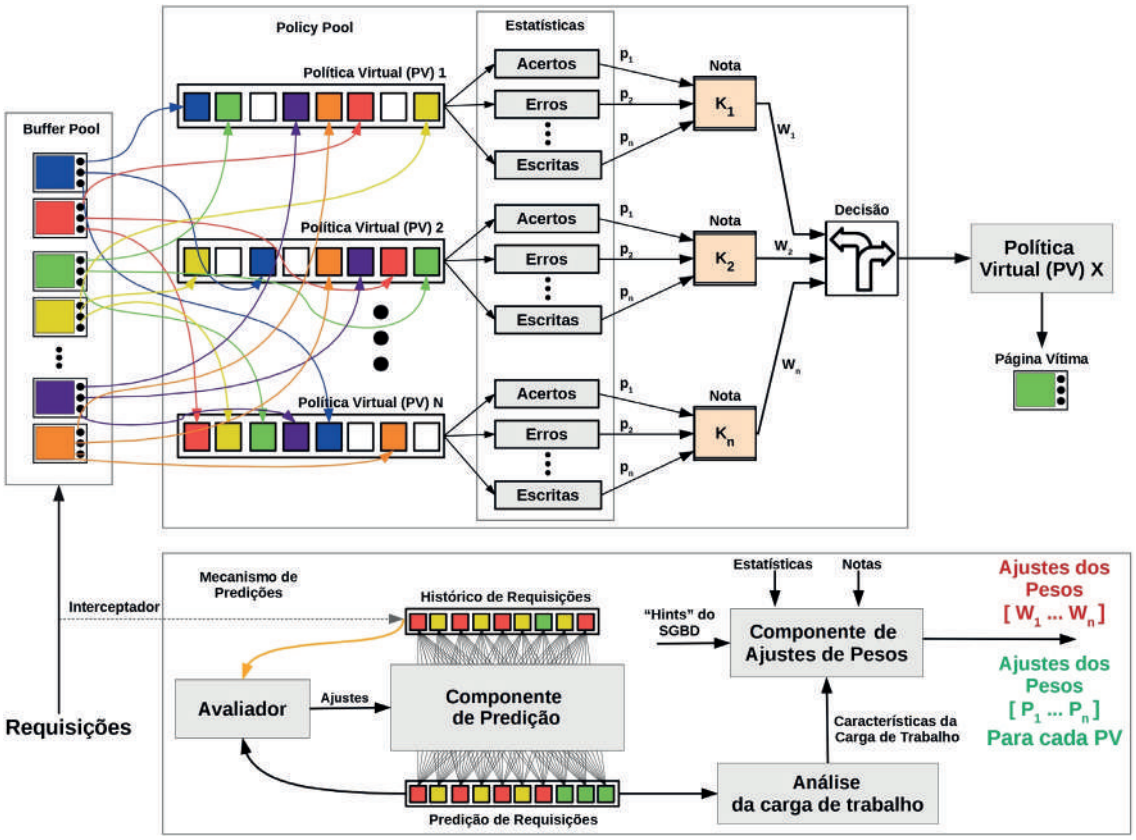


Figura 3. Estrutura da estratégia MIRP

Podemos computar metadados sobre cada PV, como por exemplo, o número de acertos (ou *hits*), erros (ou *miss*), escritas ou leituras. Com base nisso podemos determinar uma nota K a fim de avaliar o desempenho atual de cada PV. O cálculo pode ser feito de maneira ponderada pelos pesos [P_1, P_2 ... P_n] aplicados respectivamente a cada um dos metadados escolhidos, estes pesos representam o grau de importância daquele metadado. Por exemplo, uma política que pretende diminuir o número de escritas sequenciais pode ter sua nota prejudicada pelo fato dos acertos estarem muito baixos. Ao considerar os pesos, podemos ajustar os algoritmos a responderem melhor pelo seu viés de implementação, quando é aumentado o valor do peso associado para o metadado do número de escritas.

Caso a página requisitada não esteja no *buffer pool* (*miss* ou erro) porém há espaço na memória, a página é inserida no *buffer pool* e em cada uma das PV. Caso contrario, se não há espaço na memória, a estratégia toma uma decisão escolhendo uma PV X com maior nota K ponderada pelo peso W, que determina a importância daquela PV naquele momento. Dessa forma, é possível indicar qual página deve ser despejada como vítima, seguindo o seu algoritmo interno de substituição.

À medida que as requisições são invocadas, a tendência é que, com as mudanças das cargas de trabalho dinâmicas, algumas PV terão notas maiores e outras menores.

Considere um cenário em que a PV_1 se encontra ganhando, e a carga de trabalho muda para um padrão em que a PV_2 se adequaria melhor. De certa forma a estratégia deve chegar a conclusão que PV_2 é a melhor, pois irá punir a PV_1 todas as vezes que ela errar, e por outro lado irá premiar a PV_2 toda vez que ela acertar.

Um dos principais problemas do ACME é o tempo para realizar a mudança de PV. Pensando nisso, este trabalho propõe interceptar as requisições de páginas armazenando seus identificadores (*page-id*) em um histórico de requisições (uma série temporal), pois isso fornece uma visão momentânea de como está o comportamento da carga de trabalho. Poderíamos então utilizar algoritmos de predição de series temporais, como o LSTM [Hochreiter and Schmidhuber 1997] ou usar suavização exponencial [Levandoski et al. 2013] para determinar quais são as possíveis páginas que deverão ser requisitadas. Com a predição das requisições, poderíamos aplicar um algoritmo de análise da carga de trabalho para identificar qual tipo de padrão, extraindo estatísticas (características) como a taxa de intensidade de escrita ou leitura ou se a carga de trabalho é sequencial ou não. Tendo a carga de trabalho predita, o componente de ajuste de pesos realiza a combinação das informações e reajusta os pesos da *policy pool*. Estas informações são, entre outras: as estatísticas e notas atuais da *policy pool*, as últimas características extraídas da análise da carga de trabalho predita, e também pode considerar os *hints* informados pelo SGBD (*hints* são sugestões informadas em tempo de execução sobre o comportamentos dos componentes superiores ao *buffer*, como por exemplo o processador de consulta, que avisa, por exemplo, que uma leitura sequencial está sendo processada). A hipótese é que esse processo possa diminuir o tempo de ajuste que o algoritmo ACME levaria punindo ou premiando cada PV.

O componente avaliador estará constantemente verificando a qualidade das predições para ajustar dinamicamente o algoritmo de predição. Para isso, ele compara cada novo histórico de requisições com as predições feitas baseadas no histórico anterior.

6. Metodologia

O projeto de pesquisa consiste nas atividades abaixo:

1. Revisão bibliográfica sobre políticas de substituição de páginas e possíveis algoritmos de aprendizagem de máquina que podem atender ao problema.
2. Escolha dos principais algoritmos que podem atender ao problema.
3. Finalizar a especificação dos algoritmos propostos.
4. Implementação dos algoritmos, de preferência em um banco de dados comercial, por exemplo o PostgreSQL.
5. Avaliar a abordagem usando cargas de trabalho reais (*benchmarks*) e sintéticas que exploram diferentes padrões de acesso. Os testes podem comparar outras políticas usando métricas como a taxa de acerto, número de escritas e o tempo de execução semelhante a outros trabalhos [Megiddo and Modha 2003] [Jin et al. 2012].
6. Escrita da dissertação.

7. Produção de Trabalhos Científicos

Ate o momento está planejado a publicação dos seguintes trabalhos científicos:

1. Um *survey* sobre os algoritmos de substituição de páginas, atualmente em vias de submissão.

2. Artigos sobre a implementação de uma nova estratégia de substituição de páginas, utilizando técnicas de aprendizagem de máquina.

8. Conclusão

Neste trabalho, apresentamos uma noção básica sobre a implementação de um gerenciador de *buffer*, junto com alguns conceitos de meta-aprendizagem. Foi proposto uma estratégia de substituição de páginas denominado MIRP, que executa simultaneamente várias políticas de substituição, onde cada uma delas busca tratar diferentes tipos de padrões de carga de trabalho. A abordagem proposta, busca encontrar padrões de acesso futuro utilizando técnicas de aprendizagem de máquina, assim poderia indicar qual a melhor política para uma determinada situação. Ainda é necessário realizar testes e verificar se os mecanismos de predição e análise de carga de trabalho compensam serem utilizados, além disso, também identificar quais são os principais tipos de padrões de carga de trabalho e quais as políticas de substituição que melhor se ajustam a eles.

Referências

[Ari et al. 2002] Ari, I., Amer, A., Gramacy, R. B., Miller, E. L., Brandt, S. A., and Long, D. D. (2002). Acme: Adaptive caching using multiple experts. In *WDAS*.

[Effelsberg and Haerder 1984] Effelsberg, W. and Haerder, T. (1984). Principles of database buffer management. *ACM Trans. Database Syst.*

[Harizopoulos et al. 2008] Harizopoulos, S., Abadi, D. J., Madden, S., and Stonebraker, M. (2008). Oltp through the looking glass, and what we found there. In *Proceedings of the 2008 ACM SIGMOD*, SIGMOD '08.

[Hellerstein et al. 2007] Hellerstein, J. M., Stonebraker, M., and Hamilton, J. (2007). Architecture of a database system. *Found. Trends databases*.

[Hochreiter and Schmidhuber 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*.

[Jin et al. 2012] Jin, P., Ou, Y., Härder, T., and Li, Z. (2012). Ad-lru: An efficient buffer replacement algorithm for flash-based databases. *Data and Knowledge Engineering*.

[Kabra and DeWitt 1998] Kabra, N. and DeWitt, D. J. (1998). Efficient mid-query re-optimization of sub-optimal query execution plans. In *ACM SIGMOD Record*. ACM.

[Levandoski et al. 2013] Levandoski, J. J., Larson, P., and Stoica, R. (2013). Identifying hot and cold data in main-memory databases. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*.

[Littlestone and Warmuth 1994] Littlestone, N. and Warmuth, M. (1994). The weighted majority algorithm. *Information and Computation*.

[Megiddo and Modha 2003] Megiddo, N. and Modha, D. S. (2003). Arc: A self-tuning, low overhead replacement cache. In *FAST*.

[Riaz-ud Din and Kirchberg 2006] Riaz-ud Din, F. and Kirchberg, M. (2006). Acme-db: an adaptive caching mechanism using multiple experts for database buffers. In *Enterprise Information Systems VI*. Springer.

[Silberschatz et al. 2005] Silberschatz, A., Korth, H. F., and Sudarshan, S. (2005). *Database System Concepts, 5th Edition*. McGraw-Hill Book Company.

[TPC 2010] TPC (2010). BenchmarkTM C. `http://www.tpc.org/tpcc/`. Acessado em: 19/07/2019.

[Vilalta and Drissi 2002] Vilalta, R. and Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artif. Intell. Rev.*

Governança em Ecossistema de Dados

Grennda Guerra[1] Marcelo Iury S. Oliveira[1,2], Bernadette Farias Lóscio[1]

[1]Programa de Pós-Graduação em Ciências da Computação – Centro de Informática – Universidade Federal de Pernambuco (UFPE) - Recife – PE – Brasil

[2]Unidade Acadêmica de Serra Talhada – Universidade Federal Rural de Pernambuco

{gg,miso,bfl}@cin.ufpe.br

Nível: Mestrado
Admissão: Agosto de 2018
Expectativa de Conclusão: Agosto de 2020

Etapas Concluidas: Créditos em disciplinas, Definição do tema e Levantamento do Referencial Bibliográfico

Etapas Futuras: Finalizar os Critérios de Governança, Construção do *Framework*, Validação através de um Estudo de Caso e Escrita da Dissertação.

Resumo. Ecossistemas de Dados (EDs) são ambientes compostos por redes de atores autônomos que consomem, produzem ou fornecem direta ou indiretamente dados e outros recursos relacionados a dados (software, serviços e infraestrutura). Apesar de possuírem potencial para geração de benefícios, muitas iniciativas de ecossistemas têm falhado em estabelecer um gerenciamento efetivo de seus recursos e atores. Nesse contexto, torna-se cada vez mais relevante iniciativas que possam promover uma estrutura organizada na arquitetura geral dos dados. É nessa perspectiva que este trabalho buscará, a partir de ferramentas, modelos, práticas e estratégias já consolidadas no âmbito de Governança, a criação de um framework de governança que auxilie a criação, coordenação e manutenção de um Ecossistemas de Dados. E, a partir disso, espera-se melhorar a dinâmica entre os atores, otimizando suas interações e contribuindo para a redução dos desafios relacionados à publicação e consumo de dados.

Abstract. *Data Ecosystems (EDs) are environments composed of networks of autonomous actors that directly or indirectly consume, produce, or provide data and other data-related resources (software, services, and infrastructure). Although they have potential for benefit generation, many ecosystem initiatives have failed to effectively manage their resources and actors. In this context, it is becoming increasingly relevant initiatives that can promote an organized structure in the general data architecture. It is in this perspective that this work will seek, from tools, models, practices and strategies already consolidated in the scope of Governance, the creation of a governance framework that helps the creation, coordination and maintenance of a Data Ecosystems. And from this, it is expected to improve the dynamics among the actors, optimizing their interactions and contributing to the reduction of the challenges related to the publication and consumption of data.*

Palavras-chave*: Governança. Ecossistema de Dados.*

1. Introdução e Motivação

Nos constantes processos evolutivos da humanidade foram vivenciadas diversas mudanças, sobretudo no surgimento de novas formas de comunicação. Diante dos cenários de desenvolvimento desse contexto, pode-se destacar o avanço tecnológico como um dos mais significativos. As transformações ocorreram em diversas perspectivas, ocasionando alterações nos hábitos comportamentais, sociais e corporativos dos indivíduos. Estas novas interações junto ao crescimento da tecnologia permitiram, por exemplo, uma crescente produção de dados, antes restritos a suportes físicos, agora explorados de forma mais intensa, aumentando a quantidade de dados produzidos, trocados e consumidos.

Nesse contexto, torna-se relevante explorar o conceito de Ecossistemas de Dados (EDs), que podem ser definidos como ambientes compostos por redes de atores autônomos, como organizações e indivíduos, que consomem, produzem ou fornecem direta ou indiretamente dados e outros recursos relacionados a dados tais como softwares, serviços e infraestrutura [1]. No funcionamento de um EDs, encontram-se os conjuntos de dados, as fontes de dados, o software e a plataforma. E cada ator executa um ou mais papéis conectado a outros atores por meio de relacionamentos, de forma que a colaboração e a competição dos atores promovam a auto regulação de um EDs [1]. Os EDs possuem ciclos de dados, nos quais há consumidores intermediários de dados, como construtores de aplicativos e gerentes de dados, que podem compartilhar seus dados limpos, integrados e empacotados no ecossistema de forma reutilizável. Estes dados limpos e integrados são muitas vezes mais valiosos do que a fonte original [2].

Os EDs possibilitam benefícios como a criação de novas oportunidades de negócios, além de possibilitar a inovação e a criação de valor. Outro benefício esperado é facilitar o consumo e produção de dados. Por exemplo, atores que não possuem as habilidades e recursos para fornecer dados podem contratar provedores de serviços ou outros intermediários. Entretanto, embora os ecossistemas de dados tenham o potencial de gerar benefícios, muitas iniciativas têm falhado em estabelecer um gerenciamento efetivo de seus recursos e atores. De fato, enquanto o potencial do ecossistema de dados é real, a realização é mal sucedida em muitos casos. Por isso, estabelecer um EDs correto significa a coordenação adequada dos atores e o estímulo ao desenvolvimento e seu uso otimizado [1][3].

Nessa perspectiva, entende-se que a criação de estruturas de governança proporciona a possibilidade de definir uma dinâmica entre os papéis e as interações entre os membros a fim de promover uma melhor participação e engajamento [4]. Pois, compreende-se que a habilidade de criar, participar e de gerenciar EDs pode fornecer benefícios à indústria, à academia e aos governos. No que tange a esfera governamental, por exemplo, pode-se esperar como benefícios melhorias nos aspectos políticos e sociais, como na qualidade de vida, no crescimento econômico e no apoio aos processos de formulação de políticas e melhoria dos serviços ao cidadão [1]. Já nos aspectos econômicos, pode-se esperar a viabilização da inovação e a criação de novas oportunidades de negócios [1].

Desse modo, como a governança, de maneira geral, engloba um conjunto de princípios para direcionar a distribuição de deveres e direitos entre as partes interessadas [5], os seus métodos podem ser adaptados e empregados para estabelecer o nível de controle sobre uma instituição ou mesmo uma comunidade de atores para promover uma estrutura organizada na arquitetura geral dos dados. Neste contexto, esta pesquisa pretende, a partir de práticas já consolidadas, a criação de um framework de governança que auxilie a criação, coordenação e manutenção dos EDs.

Esse artigo está organizado da seguinte forma: A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta a proposta de solução. A Seção 4 descreve a metodologia de pesquisa e a Seção 5 os trabalhos futuros.

2. Trabalhos Relacionados

Podem ser encontrados vários estudos que mencionam adversidades relacionadas a dados na Web, entre elas a ausência de ferramentas adequadas para manipular os dados [3]. Destacam-se também desafios encontrados na publicação, no uso e no compartilhamento de dados, e, mais uma vez, a escassez no desenvolvimento de tecnologias ou sistemas que otimizem o ciclo dos dados [4]. É nessa perspectiva que os princípios e modelos atuais de governança podem ser adaptados para promover uma estrutura para um Ecossistema de Dados. O papel da governança estaria, assim, na definição de limitações, através de regras, critérios para a tomada de decisão, responsabilidades e limites de autonomia e ação dos participantes. Tendo como foco a utilização de estruturas para coordenar e controlar a ação conjunta dentro de um ambiente [5][6].

A Governança das Tecnologias de Informação acabou se tornando um conceito fundamental na área das TI [5]. Neste campo, surgiram e se consolidaram inúmeros guias de referências, modelos, *frameworks*, conjuntos de boas práticas e propostas de metodologias. Entre os principais existentes, evidenciam-se o COBIT (*Control Objectives for Information and Related Technologies*), que fornece um *Framework* que auxilia as organizações a alcançarem seus objetivos de forma holística englobando as diversas áreas de negócio e as TI [5][6]. Há também o ITIL (*Information Technology Infrastructure Library*) que é um conjunto de livros que abordam desde as necessidades mais estratégicas a as operacionais [6][7]. O CMM (*Capability Maturity Model*) que oferece diretrizes para o desenvolvimento de software e o PMBOK, que é um guia de melhores práticas no âmbito de gerenciamento de projetos. Entretanto, esses estudos são mais voltados a implementação da Governança em TI no âmbito organizacional [5], não abordando a Governança de dados ou mesmo de Ecossistemas de Dados como foco central.

3. Metodologia

Para alcançar o objetivo dessa pesquisa, utiliza-se uma abordagem exploratória, visto que ela proporciona maior familiaridade com o problema estudado, trazendo mais clareza para o desenvolvimento de hipóteses. Em relação aos procedimentos utilizados, para esse estudo, utiliza-se a pesquisa bibliográfica, que possui um papel fundamental na consolidação do conhecimento, como também identifica o estágio em que se encontra determinado assunto, pois a pesquisa bibliográfica abrange toda a bibliografia já tornada pública em relação ao tema [8][9]. E, após esta etapa prévia, pretende-se realizar um estudo de caso, que é um método que utiliza dados qualitativos, explorando ou descrevendo fenômenos a partir de um evento real [10]. O estudo de caso é útil para investigar novos conceitos, bem como para verificar como são aplicados e utilizados na prática elementos de uma teoria. [10]

A metodologia a ser seguida no desenvolvimento desta pesquisa baseia-se nas seguintes etapas:

- Estudo dos princípios, frameworks e modelos de governança disponíveis na literatura a partir de um levantamento bibliográfico: Nessa etapa buscou-se identificar procedimentos, ferramentas e estratégias que pudessem ser reutilizados, combinados e adaptados para a criação do framework proposto;

- Especificação de um framework de Governança para Ecossistemas de Dados;

- Validação do framework proposto através de um estudo de caso.

4. Proposta de Solução

Essa pesquisa propõe um *Framework,* que é um conjunto de técnicas, ferramentas ou conceitos que fornecem diretrizes para utilizar com eficiência recursos e processos. Basicamente, permite a criação de uma estrutura que define as formas e métodos pelos quais pode-se implementar, gerenciar e monitorar a governança de forma que auxilie no alinhamento estratégico [11].

Para o Framework proposto, inicialmente, mapeou-se de forma genérica quais seriam os elementos essenciais e os elementos secundários de um Ecossistema de Dados visando obter uma representação (fig.1).

Elementos de um Ecossistema de Dados	Elementos Secundários de um Ecossistema de Dados				
Papéis	Provedor de dados	Coordenador do Ecossistema	Provedor de Infraestrutura e Serviços	Regulamentação	Consumidor de Dados
Atores	Indivíduo	Organização Pública	Organização Privada	Terceiro Setor	
Recursos		Dados		Infraestrutura	
Padrões	Normas e Legislações				
Elementos de Contexto	Cultura	Conhecimento Técnico	Recursos Humanos	Orçamento	
Relacionamentos	Compartilhamento	Acessos	Downloads	Serviços de TI	

Figura 1: Elementos de um Ecossistema de Dados

Inicialmente, está sendo utilizado os princípios básicos de governança corporativa e o COBIT (fig. 2).

Segundo [12][5], os princípios básicos de Governança Corporativa são:

1) Transparência: que exprime a importância de cultivar a prática do desejo de informar;
2) Eqüidade: aborda a igualdade, trazendo a importância do tratamento justo e igualitário entre os grupos.
3) Prestação de contas: é um princípio que visa a prestação de contas da atuação dos agentes da governança corporativa
4) Responsabilidade corporativa: é o princípio que se preocupa com a questão de ordem social e ambiental na definição dos negócios e das operações e contempla todos os relacionamentos com a comunidade em que a sociedade atua.

E, de acordo com o COBIT, para atender aos objetivos de um negócio, as informações precisam atender os seguintes critérios: eficácia, eficiência, confidencialidade, integridade, disponibilidade, conformidade e confiabilidade.

Princípios de Governança	Elementos de um Ecossistema de Dados				
Prestação de Contas	Papéis	Atores			
Equidade	Papéis	Atores			
Transparência	Papéis	Atores	Recursos	Elementos de Contexto	Relacionamentos
Responsabilidade	Papéis	Atores	Elementos de Contexto	Relacionamentos	
Confiabilidade	Recursos	Padrões	Elementos de Contexto	Relacionamentos	
Integridade	Papéis	Atores	Recursos	Relacionamentos	
Disponibilidade	Recursos	Relacionamentos			
Eficiência	Papéis	Atores	Recursos	Relacionamentos	

Figura 2: Elementos relacionados com os princípios de Governança

Assim, cada princípio é associado e adaptado ao contexto de Ecossistema de Dados, da mesma maneira serão incluídas outras metodologias e modelos do campo da governança para a elaboração desse *Framework.*

5. Conclusão e Trabalhos Futuros

É nessa linha que essa pesquisa pretende, ao estudar o ciclo de dados destes elementos, compreender quais elementos precisam ser governados e como estes elementos se relacionam entre si. A partir disso, através dos princípios e modelos de governança, adaptá-los para esses elementos criando uma lista de princípios com conceitos no contexto de Ecossistema de Dados. Por conseguinte, espera-se formalizar o framework e validá-lo através de um estudo de caso.

Referências

[1] OLIVEIRA, Marcelo Iury S.; LÓSCIO, Bernadette Farias. What is a data ecosystem?. In: Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age. ACM, 2018. p. 74.

[2] POLLOCK, Rufus. Building the (open) data ecosystem. Open knowledge foundation Blog, v. 31, 2011.

[3] OLIVEIRA, Marcelo Iury S. et al. Towards a meta-model for data ecosystems. In: Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age. ACM, 2018. p. 72.

[4] OLIVEIRA, Lairson Emanuel RA et al. Data on the web management system: a reference model. In: Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age. ACM, 2018. p. 2

[5] LUNA, Alexandre JH de O. MAnGve: Implantando Governança Ágil. Rio de Janeiro: Brasport, 2011.

[6] PETERS, Brainard Guy. O que é Governança?. Revista do TCU, n. 127, p. 28-33, 2013.

[7] ROTH, Ana Lúcia et al. Diferenças e inter-relações dos conceitos de governança e gestão de redes horizontais de empresas: contribuições para o campo de estudos. Revista de Administração, v. 47, n. 1, p. 112-123, 2012.

[8] KÖCHE, J. C. Fundamentos de metodologia científica: teoria da ciência e prática da pesquisa. 15. ed. Petrópolis, RJ: Vozes, 1997.

[9] CERVO, A. L.; BERVIAN, P. A. Metodologia científica. 4. ed. São Paulo: Makron Books, 1996.

[10] VENTURA, Magda Maria. O estudo de caso como modalidade de pesquisa. Revista SoCERJ, v. 20, n. 5, p. 383-386, 2007.

[11] NETO, Ferreira et al. Metamodelos ontológicos de frameworks de melhores práticas de TI. 2010.

[12] DE ARRUDA, Giovana Silva; MADRUGA, Sergio Rossi; DE FREITAS JUNIOR, Ney Izaguirry. A governança corporativa e a teoria da agência em consonância com a controladoria. Revista de Administração da UFSM, v. 1, n. 1, 2008.

Um Sistema de Recomendação para Coletivos de Produtores da Agricultura Familiar

Aluno: Ivandro Claudino de Sá

E-mail: ivandro.claudino@gmail.com

Orientador: José Maria da Silva Monteiro Filho

E-mail: monteiro@dc.ufc.br

Universidade Federal do Ceará (UFC)
Programa de Mestrado e Doutorado em Ciência da Computação (MDCC)
Campus do Pici – CEP: 60.440-900 – Fortaleza – CE– Brasil

Nível: Mestrado
Ingresso: Janeiro de 2019
Conclusão prevista: Janeiro de 2021
Etapas Concluidas: Revisão Bibliográfica Preliminar, Definição do Problema.
Etapas Futuras: Qualificação, Implementação, Defesa de Dissertação.

Abstract. *Agriculture is one of the main economic activities in Brazil, responsible for much of the national Gross Domestic Product (GDP) and for the production of organic and agroecological products. Family farmers organize themselves in producer groups such as cooperatives, associations and federations. However, these groups are geographically dispersed in rural areas and usually have an autonomous administration, having, thus, great difficulty in communicating and sharing resources. This work aims to design a Recommendation System for agricultural products and crops, geared toward family farming producer groups. Through this system, groups will be able to receive suggestions of products (processed or raw) based on the production and evaluation of other producer groups, thus enabling better orientation of production, adding value to products and increasing the income of rural producers.*

Resumo. *A agricultura é uma das principais atividades econômicas no Brasil, sendo responsável por grande parte do Produto Interno Bruto (PIB) nacional e pela produção de produtos orgânicos e agroecológicos. Os trabalhadores da agricultura familiar se organizam por meio de coletivos produtores, tais como cooperativas, associações e federações. Contudo, estes coletivos estão geograficamente dispersos na zona rural e, em sua maioria, contam com uma administração autônoma. Desta forma, estes coletivos possuem uma grande dificuldade de comunicação e compartilhamento de recursos. Este trabalho tem por objetivo conceber um Sistema de Recomendação de produtos e culturas agrícolas, voltado para coletivos produtores da agricultura familiar. Por meio deste sistema, um determinado coletivo poderá receber sugestões de produtos, beneficiados ou não, com base na produção e avaliação dos demais coletivos produtores, possibilitando assim um melhor direcionamento da produção, agregando valor aos produtos e aumentando a renda dos produtores rurais.*

1. Introdução

A agricultura é uma das principais atividades econômicas no Brasil, sendo responsável por grande parte do Produto Interno Bruto (PIB) nacional. Esta atividade envolve diferentes formas produtivas, que vão desde a agricultura familiar até a monocultura agroexportadora, cada uma com suas características e especificidades.

Atualmente, a agricultura familiar é responsável pela maior parte dos empregos no campo, sendo uma das principais ferramentas de combate ao êxodo rural. A agricultura familiar contribui fortemente para a segurança alimentar e nutricional, bem como para o desenvolvimento regional e a geração de emprego e renda. Além disso, diferente da agricultura convencional, que utiliza uma variedade de produtos agrotóxicos e práticas nocivas ao meio ambiente, a agricultura familiar é hoje a maior responsável pela produção de produtos orgânicos e agroecológicos. Essa prática busca desenvolver uma nova forma de relação do ser humano com a terra, o meio ambiente e a sociedade, considerando além da produção, o melhor uso e a conservação dos recursos naturais.

Historicamente, os movimentos sociais desenvolvem um importante papel na organização dos trabalhadores rurais e no fortalecimento da agricultura familiar e da economia solidária. Por meio desses movimentos, os trabalhadores conseguem canais para se capacitarem tecnicamente e organizarem coletivos produtores, tais como: cooperativas, associações e federações.

Os coletivos da agricultura familiar produzem diferentes culturas por meio de diferentes técnicas, experiências e vivências. A produção difere também no que se refere ao beneficiamento, sendo comum tanto a venda *in natura*, como também produtos que sofreram algum tipo de processamento. Contudo, estes coletivos estão geograficamente dispersos na zona rural e, em sua maioria, contam com uma administração autônoma. De modo geral, os coletivos possuem uma grande dificuldade na comunicação, bem como no compartilhamento de informação e de recursos.

Neste sentido, este trabalho tem por objetivo conceber um Sistema de Recomendação de produtos e culturas agrícolas, voltado para coletivos produtores da agricultura familiar. Por meio deste sistema, um determinado coletivo poderá receber sugestões de produtos, beneficiados ou não, com base na produção e avaliação dos demais coletivos produtores, possibilitando assim um melhor direcionamento da produção, agregando valor aos produtos e aumentando a renda dos produtores rurais.

2. Fundamentação Teórica

2.1. Agricultura Familiar

Segundo a Organização das Nações Unidas para Agricultura e Alimentação (FAO), nove em cada dez propriedades agrícolas no mundo são geridas por famílias que produzem cerca de 80% dos alimentos no mundo e o Brasil se destaca por ser um dos maiores produtores. Possuindo atualmente 5.570 municípios além do Distrito Federal e uma população estimada em mais de 209 milhões de pessoas, a agricultura familiar é a base da economia de 90% dos municípios com até 20 mil habitantes e é responsável pela renda de 40% da população economicamente ativa do país e de mais de 70% dos

brasileiros ocupados no campo (MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO, 2018).

Fortemente relacionada com a produção de orgânicos e agroecológicos, a agricultura familiar representa mais de 75% dos produtores registrados no Cadastro Nacional de Produtores Orgânicos (CNPO), o que indicaria o reconhecimento na agroecologia e na produção orgânica uma forma de comercialização de alimentos com alto valor agregado "e que, ao mesmo tempo, são produzidos sem o uso de insumos agroquímicos, constituindo uma opção mais segura para o agricultor, para o consumidor e para o meio ambiente". (MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO, 2017).

Para garantir uma melhor comercialização dos seus produtos, os trabalhadores rurais, muitas vezes por intermédio dos movimentos sociais, buscam se organizar através de coletivos de produtores, como cooperativas, associações, ou outras formas de produção coletiva. Com isso garantem a inclusão dos agricultores familiares no sistema de produção, na coordenação da cadeia produtiva, na geração e na distribuição de renda, na prestação de serviços, nas economias em escala, nos processos de compra e venda, no acesso a novos mercados e na agregação de valor à produção, por meio de atividades de beneficiamento.

2.2. Sistemas de Recomendação

Os Sistemas de Recomendação (SR) auxiliam o processo de indicação de conteúdos, filtrando o que é irrelevante para um determinado momento e apresentando ao usuário opções próximas de seu interesse. De acordo com RICCI; ROKACH; SHAPIRA (2016) os sistemas de recomendação podem ser classificados como:

- **Content-based:** O sistema aprende a recomendar itens semelhantes aos que o usuário gostou no passado. A similaridade é calculada com base no recursos associados aos itens comparados. Técnicas clássicas de recomendação baseadas em conteúdo visam combinar os atributos do perfil do usuário em relação aos atributos dos itens.
- **Collaborative filtering**: Considerada a mais popular, a filtragem colaborativa faz recomendações para um usuário com base em itens de outros usuários com interesses parecidos.
- **Demographic**: Este tipo de sistema sugere itens baseados nos dados demográficos do usuário. As recomendações devem ser geradas considerando diferentes nichos demográficos como localização, idioma, país ou a idade do usuário.
- **Knowledge-Based:** Os sistemas baseados em conhecimento recomendam itens considerando como determinados recursos atendem às necessidades e preferências dos usuários e como o item é útil para o mesmo. Nestes sistemas, uma função de semelhança estima o quanto as necessidades do usuário correspondem às recomendações.
- **Hybrid Recommender Systems**: São sistemas de recomendação baseados na combinação de várias técnicas.

A filtragem colaborativa utiliza informações de classificação de outros usuários e itens. Este tipo de SR possibilita recomendar itens com conteúdo diferente contanto

que outros usuários tenham demonstrado interesse, reduzindo assim a superespecialização comum aos sistemas baseados em conteúdo.

As abordagens de filtragem colaborativa podem ser agrupadas em duas classes (RICCI; ROKACH; SHAPIRA, 2016):

- **Neighborhood**: As classificações de itens do usuário armazenadas no sistema são usadas diretamente para prever classificações para novos itens.
- **Model Based**: As abordagens baseadas em modelos usam as classificações de vizinhança para aprender um modelo preditivo. As características salientes de usuários e itens são capturadas por um conjunto de parâmetros do modelo, que são aprendidos com os dados de treinamento e, posteriormente, usados para prever novas classificações.

3. Trabalhos Relacionados

Na literatura é possível encontrar diversos trabalhos relacionados aos sistemas de recomendação. Barros (2013) descreve a possibilidade de implantar um SR web de conteúdos relacionados à cultura da cana-de-açúcar, baseado em regras de associação.

Elaine Venson (2002) utiliza a filtragem colaborativa para recomendações personalizadas em sistemas de comércio eletrônico. A abordagem apresentada por Elaine Venson pode ser utilizada como base para a definição e construção dos sistema de recomendação proposto nesta dissertação de mestrado, o qual tem por finalidade recomendar produtos relacionados aos coletivos de produtores rurais.

Assim como Venson (2002), FURLAN; ZAMBERLAN; VIEIRA; CANAL (2018) destacam as etapas necessárias para o processo de filtragem colaborativa e as medidas de similaridade necessárias para a construção do modelo de recomendação.

4. A Solução Proposta

Os coletivos de produtores da agricultura familiar comercializam os mais diversos produtos: frutas, grãos, legumes, mel, ovos, carnes, leite e derivados, além de polpas, farinhas, geleias, pães e até ervas medicinais, artesanato e literatura popular.

As ferramentas computacionais utilizadas por esses coletivos variam entre aplicativos de comunicação, redes sociais, até planilhas eletrônicas e sistemas de controle de estoque. São ferramentas que podem auxiliar na coleta dos dados a serem utilizados pelo sistema de recomendação proposto neste trabalho.

A solução proposta se baseará na filtragem colaborativa, usando como base de predição os produtos de outros coletivos. O sistema deverá conter um banco com os dados dos coletivos e seus respectivos itens. Esses dados serão fornecidos pelos próprios coletivos e seus membros através de uma plataforma web e um *webservice* que serão construídos. O resultado de saída será disponibilizado através da própria plataforma, que apresentará os itens recomendados de forma personalizada a cada coletivo. O processo de recomendação por filtragem colaborativa será realizado em 3 etapas: representação, construção do modelo e geração da recomendação.

A Tabela 1 ilustra um exemplo de matriz *M* x *N,* onde o *M* representa os coletivos e *N* os produtos. Nesta matriz coletivo-produto (*R),* cada elemento é assinalado com "x" se o coletivo produzir um determinado produto.

Coletivos	**Produtos**					
	Caju	**Castanha-de-caju**	**Leite**	**Ovos**	**Bolo**	**Polpa de fruta**
Coletivo 1	x	x	x			
Coletivo 2	x			x		x
Coletivo 3			x		x	

Tabela 1. Matriz com a relação de produtos por coletivo.

O objetivo do sistema de recomendação consiste em recomendar, para cada coletivo, uma lista de produtos que ainda não produz. O algoritmo utilizado deverá ponderar os vizinhos ou calcular a medida de proximidade entre os produtos. Assim, considerando os dados ilustrados na Tabela 1, pretende-se possibilitar recomendações do tipo:

- Para o coletivo 1 recomenda-se a produção de polpa-de-frutas.
- Para o coletivo 2 recomenda-se a produção de castanha-de-caju.
- Para o coletivo 3 recomenda-se a produção de ovos.

4.1. Arquitetura do Sistema

O sistema de recomendação será desenvolvido em uma plataforma web. Nesta plataforma, os coletivos da agricultura familiar e produtores realizarão o cadastro dos seus produtos e recursos que serão utilizados na construção do modelo preditivo.

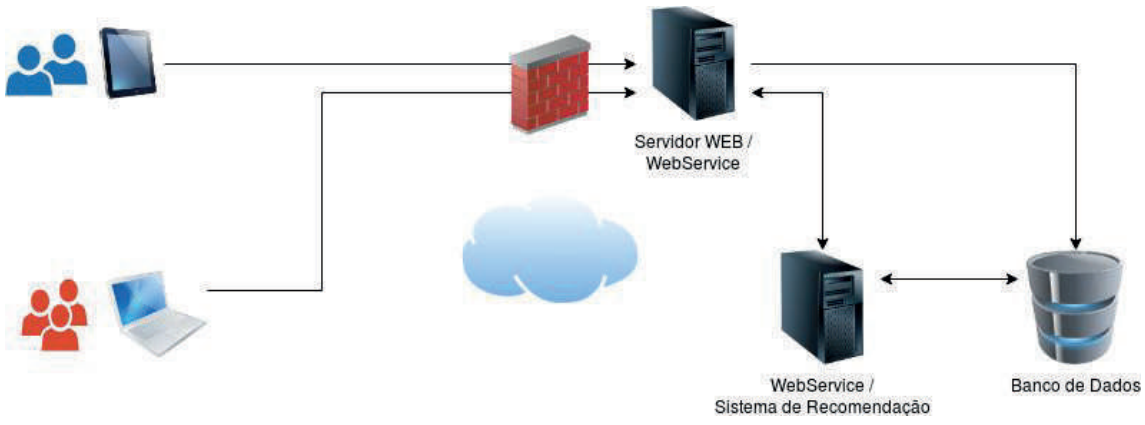


Imagem 1. Arquitetura do sistema

A arquitetura descrita na Figura 1 representa uma plataforma desenvolvida em múltiplas camadas. O *frontend* será implementado utilizando a tecnologia Angular e hospedado em um servidor Web. Esta tecnologia aplica o conceito de *Single Page Application* (SPA) e se integra a outros *frameworks* como o Ionic, para dispositivos móveis. É também no *frontend* que os produtos sugeridos pelo sistema de recomendação serão apresentados ao usuário.

Um *webservice* REST será desenvolvido como *backend* utilizando a tecnologia Java. Este *backend* avaliará as regras de negócio necessárias e garantirá as validações de segurança que permitirão adicionar, editar e excluir itens. É através dele que ocorrerá a integração com o Sistema de Recomendação.

O sistema de recomendação será desenvolvido em Python e seus recursos serão acessados somente pelo *backend*. O banco de dados do sistema será o Postgres.

5. Metodologia

Este trabalho será dividido em 4 etapas:

Na primeira etapa ocorrerá o levantamento bibliográfico e teórico, tendo como objetivo o aprofundamento do conhecimento acerca de técnicas, algoritmos e trabalhos relacionados. Nesta etapa também haverá o levantamento de informações acerca dos produtos e recursos dos coletivos de produtores da agricultura familiar junto a organizações sociais localizadas na cidade de Fortaleza - Ceará.

Na segunda etapa, será desenvolvida a plataforma para que os produtores cadastrem e mantenham as informações acerca dos produtos de seus coletivos.

Na terceira etapa, os dados serão analisados, tratados e modelos preditivos serão implementados e testados visando a construção de um modelo de recomendação.

Por fim, na quarta etapa, o sistema concebido será avaliado.

6. Conclusões

Este trabalho tem por objetivo conceber um Sistema de Recomendação de produtos e culturas agrícolas, voltado para coletivos produtores da agricultura familiar. Por meio deste sistema, um determinado coletivo poderá receber sugestões de produtos, beneficiados ou não, com base na produção e avaliação dos demais coletivos produtores, possibilitando assim um melhor direcionamento da produção, agregando valor aos produtos e aumentando a renda dos produtores rurais. Como resultado, espera-se uma ferramenta que permita aprimorar a troca de informações, tecnologias e experiências, auxiliando os coletivos de produtores da agricultura familiar em suas práticas cotidianas.

Referências

AGGARWAL, Charu C.. **Recommender Systems**: The Textbook. 1. ed. New York: Springer, 2016.

BARROS, FLAVIO MARGARITO MARTINS DE. **Um Sistema de Recomendação para Páginas Web Sobre a Cultura da Cana-de-açúcar**. 2013. Dissertação (Mestrado em

Engenharia Agrícola) - Universidade Estadual de Campinas, Campinas, 2013. Disponível em: http://repositorio.unicamp.br/bitstream/REPOSIP/256782/1/Barros_FlavioMargaritoMartinsde_M.pdf. Acesso em: 14 jul. 2019.

BARROS, Flávio Margarito Martins de; OLIVEIRA, Stanley Robson de Medeiros; OLIVEIRA, Leandro Henrique Mendonça de. **Desenvolvimento e validação de um sistema de recomendação de informações tecnológicas sobre cana-de-açúcar**. Bragantia, Campinas, v.72, n.4, p.287-395, 2013. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0006-87052013000400010&lng=en&nrm=iso. Acesso em 18 jul. 2019.

FURLAN, Leonardo Antônio da Rosa; ZAMBERLAN, Alexandre de Oliveira; VIEIRA, Sylvio André Garcia; CANAL, Ana Paula. **Desenvolvimento de um Sistema de Recomendação Para Bibliotecas Digitais**. Disciplinarum Scientia, Santa Maria, 2018. Disponível em: https://periodicos.ufn.edu.br/index.php/disciplinarumNT/article/view/2591. Acesso em: 14 jul. 2019.

MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO. Secretaria de Agricultura Familiar e Cooperativismo. **Agricultura familiar do Brasil é 8ª maior produtora de alimentos do mundo**. Disponível em: http://www.mda.gov.br/sitemda/noticias/agricultura-familiar-do-brasil-%C3%A9-8%C2%AA-maior-produtora-de-alimentos-do-mundo. Acesso em: 1 jul. 2019.

MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO. Secretaria de Agricultura Familiar e Cooperativismo. **Brasil: 70% dos alimentos que vão à mesa dos brasileiros são da agricultura familiar**. Disponível em: http://www.mda.gov.br/sitemda/noticias/brasil-70-dos-alimentos-que-v%C3%A3o-%C3%A0-mesa-dos-brasileiros-s%C3%A3o-da-agricultura-familiar. Acesso em: 1 jul. 2019.

MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO. Secretaria de Agricultura Familiar e Cooperativismo. **Mais orgânicos na mesa do brasileiro em 2017**. Disponível em: http://www.mda.gov.br/sitemda/noticias/mais-org%C3%A2nicos-na-mesa-do-brasileiro-em-2017. Acesso em: 1 jul. 2019.

MINISTÉRIO DO DESENVOLVIMENTO SOCIAL. Secretaria Nacional de Segurança Alimentar e Nutricional. **Catálogo de produtos ofertados pela agricultura familiar**. Brasília: [*s. n.*], 2018. Disponível em: http://www.mds.gov.br/webarquivos/arquivo/seguranca_alimentar/Simposio_PAA/SIMPOSIO_NACIONAL/Catalogo_Produtos_Agricultura_Familiar.pdf. Acesso em: 1 jul. 2019.

RICCI, Francesco; ROKACH, Lior; SHAPIRA, Bracha. **Recommender Systems**: Handbook. 2. ed. New York: Springer, 2015.

VENSON, Elaine. **Um Modelo de Sistema de Recomendação Baseado em Filtragem Colaborativa e Correlação de Itens para Personalização no Comércio Eletrônico**. 2002. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis, 2002. Disponível em: https://repositorio.ufsc.br/handle/123456789/83047. Acesso em: 8 jul. 2019.

Uma Estrutura de Indexação para Eventos de Trânsito

Mariana Machado Garcez Duarte[1]
Orientadora: Carmem Satie Hara[1]
Coorientadora: Rebeca Schroeder Freitas[2]

[1] Departamento de Informática – Universidade Federal do Paraná
Caixa Postal 19.081 – 81.531-990 – Curitiba, PR – Brasil

[2]Centro de Ciências Tecnológicas – Universidade Estadual de Santa Catarina
Joinville - SC - Brazil
{mmgduarte,carmem}@inf.ufpr.br, rebeca.schroeder@udesc.br

Nível: Mestrado
Ingresso no Programa: Março/2018
Previsão de Conclusão: Março/2020
Etapas Concluídas: Defesa da qualificação Maio/2019
Etapas Futuras: Desenvolvimento, Análise, Defesa da dissertação

Abstract. *In the urban environment, data collected from traffic events can serve as elements of study for city planning. Because of the speed data is reported, each event is usually stored as an individual record. Although this approach guarantees fast insertions in a database, it produces poor performance for queries to recover events that satisfy spatio-temporal filters. To address this problem, this paper proposes a method for indexing mobility data (MIDM), that combines a B+ tree, to partition data based on their temporal dimension, with a bitmap index to address their spatial dimension. The goal of MIDM is to achieve low cost insertion operations, along with good performance spatial-temporal queries.*

Resumo. *Dados coletados a partir de eventos no trânsito, como engarrafamentos, acidentes e enchentes são importantes para o planejamento da mobilidade em cidades. O armazenamento de eventos pode ser feito na forma de registros individuais em razão da velocidade com que estes dados são reportados. No entanto, consultas que necessitam recuperar eventos baseados em filtros espaço-temporais tem baixo desempenho neste modelo de armazenamento. Para tratar este problema, este artigo propõe um Método para a Indexação de Dados de Mobilidade (MIDM), que combina uma árvore B+ para particionar os registros pela dimensão temporal e índices bitmap associados à técnica de tesselação para tratar a dimensão espacial. O objetivo do MIDM é garantir um baixo custo de inserção, bem como bom desempenho no processamento de consultas espaço-temporais.*

Palavras-chave: *busca espaço-temporal, indexação espaço-temporal, eventos no trânsito, índices bitmap*

1. Introdução

Dados coletados a partir de eventos no trânsito, como engarrafamentos, acidentes e enchentes são importantes para o planejamento da mobilidade em cidades. Em geral, eventos são armazenados como registros individuais, em razão da velocidade com que eles são reportados. No entanto, consultas que recuperam eventos baseados em filtros espaço-temporais tem baixo desempenho neste modelo de armazenamento. Para tratar este problema, o artigo propõe o **M**étodo de **I**ndexação de **D**ados de **M**obilidade (MIDM). O objetivo do MIDM é aliar um baixo custo de inserção a um bom desempenho no processamento de consultas espaço-temporais.

O MIDM contempla três estratégias: indexação espacial, indexação temporal e associação de objetos à indexação espaço-temporal. Para a indexação espacial, a estratégia baseia-se na técnica de tesselação, na qual a área geográfica de interesse é dividida em uma grade, formando células geográficas (CGs), como ilustrado na Figura 1. Dessa forma, eventos que ocorrem na mesma CG são agrupados em blocos, garantindo assim que sejam armazenados de forma contínua no disco. Eventos possuem intrinsecamente uma dimensão temporal. Como o período de monitoramento dos eventos pode ser longo, o MIDM divide os blocos por períodos de tempo e cria um índice B+ sobre estes períodos. A estratégia de associação de objetos à indexação espaço-temporal é baseada em índices bitmap. Ruas, bairros e pontos de interesse, como parques e escolas, são exemplos de objetos no contexto urbano. Tais objetos podem ser considerados como chave de um índice bitmap, no qual o vetor de bits é indexado pelas células geográficas, como ilustrado na Figura 4.

O restante do artigo está organizado da seguinte forma: A Seção 2 apresenta o método proposto. A Seção 3 apresenta trabalhos relacionados. A Seção 4 apresenta a metodologia. A Seção 5 apresenta o cronograma. A Seção 6 apresenta a conclusão e trabalhos futuros.

2. O Método MIDM

O método proposto neste trabalho tem como objetivo reduzir o custo computacional de consultas espaço-temporais. O diferencial da proposta é a combinação da utilização de bitmaps e a distribuição dos registros pela informação espaço-temporal. Na Figura 1, a arquitetura da indexação espacial do método é mostrada. O carregamento da estrutura do MIDM consiste em três fases: aplicação da técnica de tesselação sobre a área de interesse, inserção dos registros de eventos em blocos e a indexação com uma árvore B+ e bitmaps. Ao fim do carregamento da estrutura, é possível realizar consultas.

2.1. Criação e Inserção

As entradas para a criação da estrutura do MIDM e inserção dos dados são: a base de dados, a área de interesse, tamanho da grade geográfica, o tamanho dos blocos de registros e a quantidade de níveis da árvore temporal (por exemplo horas, dias, semanas, ou meses). Na primeira fase do carregamento, é utilizada a técnica de tesselação, na qual é realizada a divisão geográfica da área de interesse, na forma de grade, dividindo-a em células geográficas (CGs) de tamanho homogêneo (Figura 2b).

Na segunda fase do carregamento é feita a leitura da base original. Os registros possuem um *timestamp* (Ts) e um conjunto de pares de latitude (Lat) e longitude (Long).

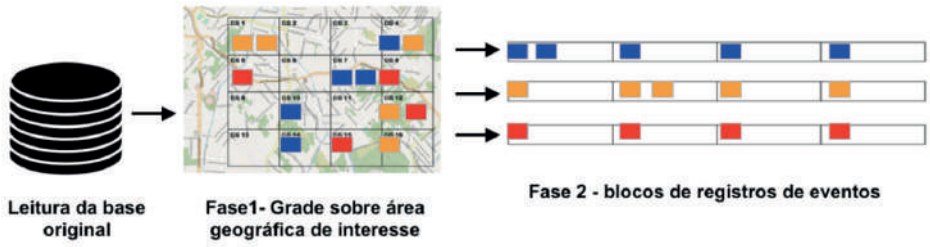
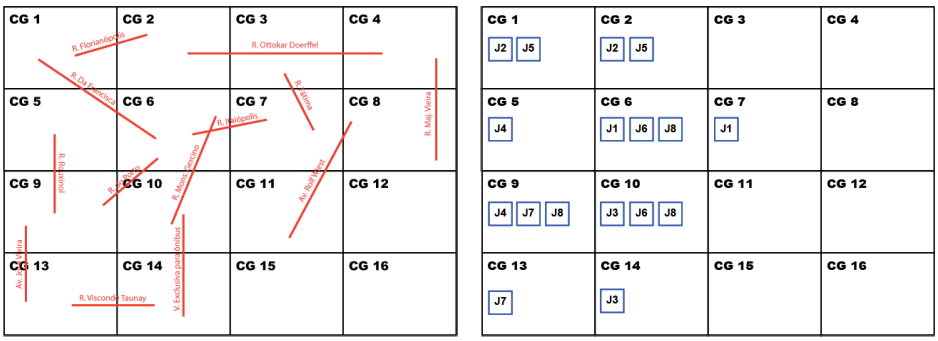


Figura 1. Parte Geográfica do Método Proposto

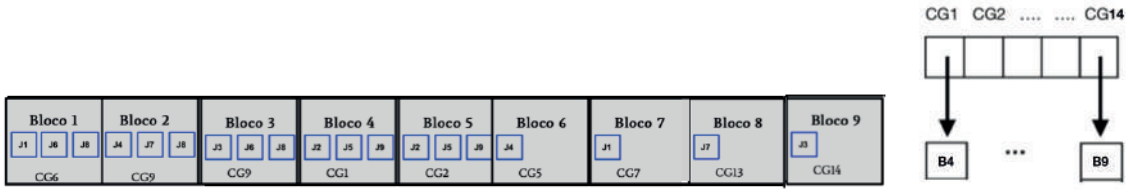


(a) Grade de CGs com ruas com engarrafamento

(b) Inserção de registros em CGs

Figura 2. Ruas e eventos separados por CGs

Assim, é feita a distribuição, separando os registros de eventos por CGs, para cada intervalo de tempo definido e cada classe de evento (engarrafamento, alerta e irregularidade) (Figura 2b). Quando a quantidade de registros em uma CG ultrapassar o limite pré-definido para o tamanho do bloco de registros, este conjunto de registros é armazenado em disco, em um bloco que possui referência à CG correspondente, juntamente com o período inicial do primeiro registro e o final do último registro nele armazenado. A Figura 3a ilustra esta organização. Quando a leitura é encerrada os blocos restantes também são gravados em disco. Os blocos têm tamanho fixo e cada classe de evento gera um arquivo separado. A fim de obter todos os blocos que se referem à mesma CG, cada arquivo é associado a um vetor de blocos (Figura 3b). As posições do vetor correspondem às CGs e cada elemento do vetor está associado a uma lista de blocos. Tais blocos possuem registros cuja localização geográfica pertença à CG correspondente. Na ilustração, o bloco 4 possui registros de eventos que ocorreram na célula CG1 e o bloco 9 possui registros da CG14.



(a) Organização do Arquivo em blocos

(b) Vetor de Blocos

Figura 3. Arquivo e Vetor de Blocos

2.2. Indexação

Para a indexação, na terceira fase do carregamento é criada a árvore temporal, que é uma árvore B+, com os intervalos de tempo definidos na segunda fase. Os nós folha da árvore temporal correspondem à menor granularidade de intervalo de tempo definida pelo usuário, como por exemplo, uma hora. Cada folha é associada a vetores de bloco contendo eventos ocorridos no período de tempo correspondente, bem como índices bitmap (Figura 4).

Os índices bitmap são criados a partir dos registros de eventos e da geometria dos elementos da aplicação que são a chave de busca do índice. Por exemplo, a Figura 2a ilustra um exemplo da classe engarrafamento com a geometria das ruas. A Figura 4 representa uma árvore temporal com um índice bitmap em uma de suas folhas. O índice é sobre o domínio das ruas r=r_1,..., r_n que tiveram relato de engarrafamento, onde os vetores de bitmap vBit[ri] são indexados pelas CGs. Caso a rua pertença a uma CG e exista um engarrafamento, é inserido o bit 1, caso contrário o bit é 0. Os vetores de bits dos índices bitmap sempre são indexados sobre CGs para que seja possível utilizar as operações tradicionais de índice bitmap, como conjunção e disjunção.

2.3. Consulta

Ao fim do carregamento da estrutura, é possível realizar consultas. O processamento de uma consulta espaço-temporal inicia percorrendo a árvore temporal para obter os vetores de bloco e índices bitmap que possuem registros no período da consulta. A partir dos índices bitmap são definidas as CGs que possuem valores para o resultado e a partir dos vetores de blocos, os blocos das CGs que sobrepõem a área geográfica da consulta são recuperados para que os os registros que satisfazem aos filtros da consulta componham o resultado.

A geração de blocos nas folhas da árvore temporal permite estender futuramente o MIDM para o processamento paralelo dos blocos que potencialmente produzem resultados para a consulta. O MIDM também tem a possibilidade de ser estendido futuramente para o processamento de dados em fluxo já que a área de interesse é dividida em uma grade geográfica e a distribuição dos registros em blocos é baseada somente na sua localização geográfica.

3. Trabalhos Relacionados

A árvore STIG [Doraiswamy et al. 2016] é proposta para otimizar consultas espaço-temporais utilizando GPUs. Ela corresponde a uma árvore K-D na qual as folhas consistem de um conjunto de registros, chamado de bloco. O aspecto temporal é uma das dimensões da árvore. A ideia de utilizar blocos é permitir que eles sejam processados em paralelo na execução temporal de uma consulta.

O modelo de indexação de [Imawan et al. 2015] tem a proposta de prever congestionamentos de veículos a partir do índice TiQ, que leva em consideração as características espaço-temporais de dados de trânsito. O modelo é otimizado para pesquisas analíticas com linha de tempo. Para encontrar padrões do tráfego diário em estruturas rodoviárias, o índice TiQ mantém dois componentes principais: um índice de localização e um índice de tempo. O índice de localização representa a rede de ruas e conecta a visão de linha do tempo dos engarrafamentos com os segmentos das ruas. Este índice é baseado em hash e

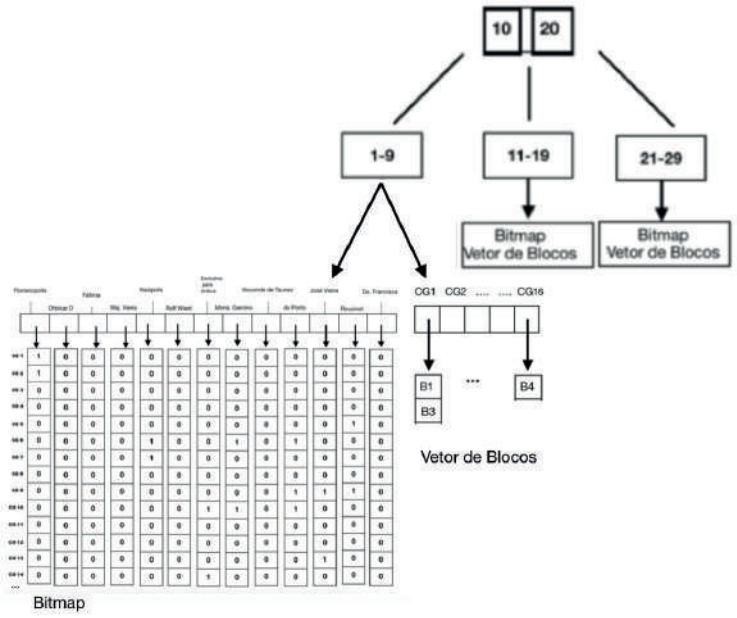


Figura 4. Árvore Temporal com períodos de tempo, bitmaps e vetores de blocos

tem como chave o identificador do segmento da rua e a tripla: identificador da próximo segmento de rua, identificador do segmento de rua anterior e um conjunto de apontadores para linhas do índice temporal.

A Árvore Buneman [Buneman et al. 2004] é proposta para realizar o versionamento de uma base de dados, trabalhando com dados focados em objeto e com o aspecto temporal. Um conjunto de experimentos demonstra que o arquivo não gera sobrecarga de espaço de armazenamento.

O trabalho de [Keawpibal et al. 2012] apresentou o algoritmo E-EBI para otimizar buscas de igualdade em bitmap codificado. Ele usa o menor número de vetores bitmap e menor tempo de consulta comparado a outras técnicas de indexação de bitmap existentes, usando operações booleanas de baixo custo. Em [Keawpibal et al. 2016] é apresentado um índice bitmap de codificação, chamado *HyBiX*, para a codificação híbrida de índices bitmap. Os resultados experimentais mostram que o índice pode reduzir o requisito de espaço com atributos de cardinalidade altos e com tempos de execução satisfatórios para consultas de igualdade e intervalo.

O índice SB [Doraiswamy et al. 2016] é proposto para Data Warehouses (DW) espaciais e suporta hierarquias de atributos espaciais pré-definidas, bem como lida com a multidimensionalidade. É utilizada a implementação de bitmap chamada FastBit, juntamente com o índice HSB, no qual é possível realizar a escolha entre estes dois índices. O HSB é um índice espacial baseado em árvore, com cada nodo folha apontando para um vetor de bits de um índice bitmap de junção estrela. Consultas de baixa seletividade beneficiam-se com o índice HSB e o índice SB é indicado para consultas seletivas.

A Tabela 1 relaciona os trabalhos relacionados pelos aspectos temporais, geográficos, se é possível realizar a indexação de objetos e se utilizam bitmaps na estrutura do índice. O método proposto combina partes dos métodos mencionados anteriormente para otimizar tanto aspectos temporais como geográficos para consultas espaço-

Estrutura	Temporal	Geográfico	Indexação de objetos	Bitmap
Árvore STIG	X	X		
Índice TiQ	X	X		
Árvore Buneman	X		X	
Enhanced Encoded Bitmap				X
Índice HyBiX bitmap				X
Índice SB		X		X
Índice HSB		X		X
MIDM	X	X	X	X

Tabela 1. Comparação entre os Trabalhos Relacionados

temporais e a utilização de bitmaps para diminuir o espaço de busca. A árvore STIG [Doraiswamy et al. 2016] orientou a criação de blocos de registros no método proposto. A árvore Buneman [Buneman et al. 2004] orientou a criação da árvore temporal. Os trabalhos de [Keawpibal et al. 2012] e [Keawpibal et al. 2016] orientaram a utilização de técnicas de compressão de bitmap. Os índices SB e HSB [Siqueira et al. 2012] orientaram a utilização de índices bitmap.

4. Metodologia

O MIDM possui aplicação em diversos contextos, porém será utilizada a base do aplicativo Waze[1], resultante do trabalho em conjunto com a prefeitura de Joinville-SC. A base possui 16 GB, com dados do período de setembro 2017 a setembro 2018. Os critérios de avaliação do desempenho a serem considerados neste trabalho incluem: complexidade dos algoritmos, número de páginas transferidas para a memória, tempo de construção e carregamento da estrutura de indexação, tempo de execução de consultas e espaço necessário para armazenamento.

Será avaliado o desempenho o método durante as consultas com volume de dados e volume de requisições (*throughput*) crescentes. Os parâmetros para teste serão o tamanho da área de interesse, o tamanho da grade geográfica, tamanho do bloco e quantidade de níveis da árvore temporal. O método será implementado na linguagem C e para a implementação do bitmap será utilizado CRoaring [2] com base na recomendação de [Wang et al. 2017].

Algumas consultas foram definidas para avaliação do MIDM. A seguir as consultas são apresentadas: **1-** Em quais dias da semana houve engarrafamento no primeiro trimestre de 2018 na zona central? **2-** A rua R apresentou no instante I engarrafamento? **3-** Na quadra do hospital H, houve engarrafamento no ano de 2018? **4-** Quais bairros apresentam engarrafamento no horário H? **5-** Qual é a quantidade de alagamentos no bairro B? **6-** Qual rua teve engarrafamento e alerta? **7-** Após a implementação do corredor de ônibus, a média de engarrafamento do último mês aumentou? As consultas foram baseadas na taxonomia definida por [Vaisman and Zimányi 2009]. A consulta 1, 2,3 e 4 são baseadas na classe TOLAP Espacial. Já, as consultas 5 e 6 são definidas pelo modelo SOLAP. Por fim, a consulta 7 é baseada no padrão TOLAP Espaço Temporal.

5. Cronograma

As etapas já concluídas da dissertação são:

[1]https://github.com/joinvalle/Joinville-Smart-Mobility
[2]https://github.com/RoaringBitmap/CRoaring

	6/2019	7/2019	8/2019	9/2019	10/2019	11/2019	12/2019	1/2020	2/2020
Escrita de Dissertação	X	X	X	X	X	X	X	X	X
Implementação	X	X	X	X	X				
Experimentos			X	X	X	X			
Análise dos Resultados			X	X	X	X	X		
Escrita de Artigos	X	X	X	X	X	X	X	X	X
Defesa									X

Tabela 2. Planejamento

1. Levantamento Bibliográfico
2. Definição de Trabalhos Relacionados
3. Definição de Consultas
4. Defesa da Proposta

Como etapas futuras serão realizados os passos apresentados na Tabela 2.

6. Considerações

Este artigo apresentou uma proposta do método MIDM, para o armazenamento e indexação de eventos de trânsito. O diferencial da proposta é a combinação de uma árvore B+ para particionar os registros pela dimensão temporal e índices bitmap associados à técnica de tesselação para tratar a dimensão espacial. O objetivo do MIDM é garantir um baixo custo de inserção, bem como bom desempenho no processamento de consultas espaço-temporais. O método está em processo de implementação.

Referências

Buneman, P., Khanna, S., Tajima, K., and Wang-Chiew (2004). Archiving Scientific Data. *ACM Transactions on Database Systems (TODS)*, 29(1):2–42.

Doraiswamy, H., Vo, H. T., Silva, C. T., and Freire, J. (2016). A GPU-Based Index to Support Interactive Spatio-Temporal Queries over Historical Data. In *IEEE 32nd International Conference on Data Engineering (ICDE)*, Helsinki, Finland.

Imawan, A., Putri, F., and Kwon, J. (2015). TiQ: A Timeline query processing system over Road Traffic Data. In *IEEE International Conference on Smart City*, Chengdu, Ch.

Keawpibal, A., Wattanakitrungroj, N., and Vanichayobon, S. (2012). Enhanced Encoded Bitmap Index for Equality Query. In *2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*, Seoul, South Korea.

Keawpibal, N., Preechaveerakul, L., and Vanichayobon, S. (2016). HyBiX: A novel encoding bitmap index for space-an time-eficient query processing. Songkhla, Thailand.

Siqueira, T. L. L., de Aguiar Ciferri, C. D., Times, V. C., and Ciferri, R. R. (2012). The SB-index and the HSB-index: efficient indices for spatial data warehouses. *Geoinformatica*, 16(1):165–205.

Vaisman, A. and Zimányi, E. (2009). What is spatio-temporal data warehousing? In *Proceedings of the 11th International Conference on DW and Knowledge Discovery*, DaWaK '09, pages 9–23, Berlin, Heidelberg. Springer-Verlag.

Wang, J., Lin, C., Papakonstantinou, Y., and Swanson, S. (2017). An Experimental Study of Bitmap Compression vs. Inverted List Compression. In *SIGMOD*, Chicago, IL, USA.

Predicting Music Success by Combining Song Features and Social Metrics

Mariana O. Silva[1], **Mirella M. Moro**[1]

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

{mariana.santos,mirella}@dcc.ufmg.br

Nível: Mestrado
Data de ingresso no mestrado: Março de 2018
Data prevista para conclusão do mestrado: Março de 2020
Etapas concluídas: Revisão da literatura;
Definição do problema; Coleta de dados;
Caracterização do banco de dados de músicas;
Seleção dos principais atributos e principais interações;
Estudo de causalidade entre os fatores de sucesso musical.
Publicações:
[Silva et al. 2019]
[Silva and Moro 2019]

Resumo. *Na indústria musical, tornar-se bem-sucedido não é trivial, mas pode gerar milhões em receita. Por esse motivo, a capacidade de prever a popularidade das músicas oferece enormes benefícios para muitos domínios e públicos-alvo. Essa habilidade é a principal motivação por trás do campo de pesquisa emergente chamado de Hit Song Science, com o objetivo de prever a popularidade das faixas musicais, conforme apresentado nos top charts. Em um contexto tão relevante, este estudo propõe desenvolver um método de predição de popularidade automática para músicas antes que elas sejam lançadas. No geral, planejamos explorar métodos de aprendizado de máquina para combinar vários atributos de uma música (por exemplo, melodia, harmonia, ritmo, conteúdo de letras, contexto social, reputação do artista e perfis de colaboração musical) para prever se outras músicas subirão para uma posição alta nos charts.*

Abstract. *In the music industry, becoming successful is not trivial, but can lead to millions in revenue. For this reason, the capacity to predict the popularity of songs provides tremendous benefits for many domains and audiences. This ability is the main drive behind the emerging research field referred to as Hit Song Science, aiming in predicting the popularity of musical tracks, as presented in top charts. In such a relevant context, this study proposes to develop an automatic popularity prediction method for songs before they are released. Overall, we plan to explore machine learning methods to combine multiple attributes of a song (e.g., melody, harmony, rhythm, lyrics content, social context, artist's reputation and musical collaboration profiles) towards predicting whether other songs will rise to a high position in the charts.*

1. Introduction

The fast evolution in technology continues to drive changes in the way people discover and engage with music content. In 2018, the U.S. music industry experienced its third year of consecutive growth with retail revenues up 12% to $9.8 billion. The double-digit increase was accelerated primarily by raised revenues from paid subscription services including Spotify, Tidal and others. Specifically, streaming currently comprises 75% of total industry revenues.[1]. In such a huge industry, becoming successful is not trivial. From the musician to the producer, they are completely at the mercy of a fickle public. When releasing a single, professionals experience a complex task in trying to please everyone. No use to bet all the chips on a song if the public does not bite. Besides, some hit songs become acknowledged masterpieces, while other hit songs fall into oblivion as so-called *one-hit-wonders*. Such information implies an intriguing question: *what are the reasons for a song to achieve success and keep it for so long?* Discovering such reasons may lead to predicting whether a song will become successful, increase sales of physical and digital albums, improve the billing of on-demand audio streams services, or even help to predict the next music star.

The ability to predict the popularity of songs provides tremendous benefits for many domains and audiences. For the music industry CEOs, it may help to maximize future success by helping to choose in whom to invest to produce potential big hits. Furthermore, by investing correctly in potential artist/song and its distribution, the studio could increase sales of physical and digital albums, improve the billing of on-demand audio streams services, or even launch the next *it* artist or *summer hit*. For the music artist, it may help to choose that one song to lead the album to early stardom. For the music consumers, it may help to decide if an album is worth buying because it may potentially contain 3-5 hits, instead of being an *one-hit-wonders* album. This ability is the main drive behind the research field referred to as *Hit Song Science* (HSS), which Pachet and Sony define as "an emerging field of science that aims at predicting the success of songs before they are released on the market" [Pachet and Sony 2012].

In such a relevant context, this study aims at developing an automatic popularity prediction method for songs before they are released. To do so, we plan to explore machine learning methods to combine multiple attributes of a song. Our premise is: music is a multifaceted item, and each of its facets may be mapped to a popularity aspect (e.g., melody, harmony, rhythm, lyrics content, social context, artist's reputation and musical collaboration profiles); then each aspect can be studied, compared by statistical methods, and analyzed by an automated learner for a final decision on the popularity of a particular song. Overall, the goal is to explore the popularity of hit songs and analyze their attributes towards predicting whether other songs will rise to a high position in the charts.

Motivation. The premise of HSS is that popular songs are similar to the set of attributes that make them appealing to most people. Those attributes could then be explored through Machine Learning techniques in order to predict whether a song will go to the top position in the popularity charts. It is widely claimed that the broad range of characteristics that lead to the popularity of a song exceeds its intrinsic content, namely the audio features and the lyrics. Factors such as acoustic and lyrical features [Dhanaraj and Logan 2005], associated video clip of the track [Chon et al. 2006],

[1]RIAA 2018 Year-End Music Report, (July 28, 2019), `https://bit.ly/2JYta0x`

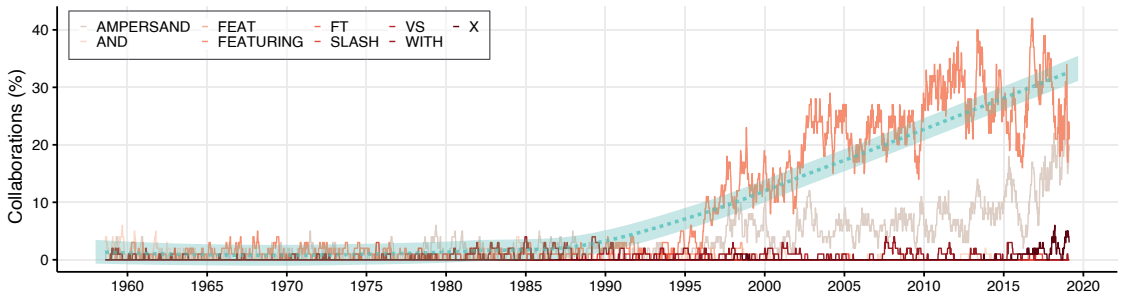


Figure 1. Billboard Hot 100 songs (1958 - 2019). Solid lines represent different types of musical collaborations. The dashed line indicates the trend smoothed line of all collaborations.

the artist's preferred attachment [Bischoff et al. 2009], social metrics [Ren et al. 2016] or social media data [Araujo et al. 2017] can also fulfill a key role.

Even with such diverse background, existing research in the area recognizes that there are alternative factors that have not yet been addressed. For instance, how artists connect professionally could be one of the aspects that significantly affect musical success. Artist collaboration is frequently thought as one of the vastest creative forces driving music today. Thanks to digital media, a broader variety of musical collaborations are emerging among artists who can often live on different continents. Analyzing data from the Billboard Hot 100 (Figure 1), before the 90s, collaborations were relatively rare (about 5% of charted songs) and generally took the form of duets. The boom in collaborations started in the mid-1990s, when the number of collaborations increased significantly, with the duets dying and the artists on display taking over.

Despite the propagation of musical collaborations, factors that lead to the success of a collaborative process are not entirely understood [Iaffaldano 2018]. Previous research in music collaboration networks has established that social interaction can influence the trajectory of the music industry [Ren et al. 2016, Araujo et al. 2017, Calefato et al. 2018]. In particular, Budner and Grahl found that different collaborative roles hold a distinct impact on network connectivity [Budner and Grahl 2016]. Furthermore, the influence of a specific function depends not exclusively on the number of relationships but also on the importance of these connections. With a sounder understanding of how the way artists connect professionally can affect their musical success, we can develop alternative tools and practices that help the music industry and its members flourish.

Research Goals. Music involves many features related to the composition (melody, harmony, rhythm or lyrics) and the social context (reachability and style of the artist, profiles of musical collaboration, culture, period, and so forth). Likewise, a person's musical preference can be influenced by many factors such as music composition, personality and his/her spatial and temporal contexts. With so many variables at hand, predicting song popularity represents a complex task that requires to find a balance between features of songs, artists and listeners. Therefore, our specific Research Issues (RI) are:

1. Identify the (potentially) intrinsic features and indicators that influence listeners and the popularity of both songs and artists;

2. Investigate the impact of these features on popularity over time. That is, dynamically analyze whether features affect the popularity level of an artist/song;
3. Further our understanding of the causal relationship between collaborative profiles and musical success;
4. Propose a machine learning approach to derive a song's popularity based on these groups of features and determine the best way for combining them to predict the success of a song.

2. Related Work

Over the years, the number of artists and musical productions has considerably increased; and so have the number of attempts to discover the recipe for turning a song into a hit. Indeed, there are plenty of analyses on factors that potentially influence musical success from varied perspectives. These analyses are part of the emerging field of science known as *Hit Song Science* (HSS). HSS has emerged as a field of predictive studies to better understand the relation between the intrinsic characteristics of songs and their popularity. In this context, popularity is regarded as a feature of a song, and the problem then is to map this feature to other resources that can be measured objectively [Pachet and Sony 2012]. Specifically, the related works follow two main directions: extracting general acoustic lyrics attributes from songs or approaches considering more subjective and social attributes related to songs popularity.

Acoustic and Lyrics Features. In one of the earliest studies in the field, Dhanaraj and Logan extracted acoustic as well as lyric-based features and then use standard classifiers to separate hit songs from non-hits [Dhanaraj and Logan 2005]. They show that lyrics-based attributes are more useful than acoustic attributes for correctly identifying hits. In more recent work, Lee and Lee use data from Billboard charts only for Rock music [Lee and Lee 2015]. Moreover, they analyze the song complexity based on the chroma, rhythm, timbre and arousal complexity features extracted from the music signal and the early stage popularity. The classification experiment for the data of the Billboard Rock Songs Chart, over about five years, show that the two groups of features (i.e., complexity and early stage popularity) are effective for different popularity patterns and combining the two types of features can be synergetic. In the same year, Singhi and Brown propose song lyrics and audio content features for predicting hits [Singhi and Brown 2015]. As in the work described earlier [Lee and Lee 2015], their data set is also derived from Billboard with varying definitions about the notion of popularity. Regarding the results, the authors conclude that a combination of lyrics and audio features performs better in identifying hits, although only lyrics resources are more useful in separating hits from non-hits.

Social Influence. Previous knowledge of a song's success or about a community's preferences can influence the musical taste of listeners. This was exactly the phenomenon studied by Salganik et al. through an impressive experiment [Salganik et al. 2006]. In the study, the authors create an artificial "music market", where $14,341$ participants downloaded unknown songs with or without knowledge of the choices of previous participants. The conclusions confirm the hypothesis that social influence contributes both to inequality and unpredictability in cultural markets. Focusing on social networks, Kim et al. propose to collect the behavior of Twitter users-listeners based on *hashtags* related to songs for

predicting popularity rankings [Kim et al. 2014]. The reported results show high correlations between behavior in listening to music by Twitter users and the trend of song popularity in general. Following a similar approach, Ren et al. predict the popularity of a song by focusing on the social network Last.fm. Their main findings indicate that the content of the music is an important determinant of a music track's time duration in terms of weeks of popularity online [Ren et al. 2016]. On a broader perspective, explaining or predicting the success of creative individuals through social network analysis has been a hot topic for decades. For instance, Calefato et al. measure creative collaboration in a music community where authors compose songs together through overdubbing [Calefato et al. 2018]. The authors evaluate the relationship between metrics related to the song- and author-related measures and the likelihood of a song being overdubbed.

The studies considering acoustic attributes and/or lyrics content disregard how external factors influence the popularity of a song. For example, the reputation of the artist/album, the social influence and context or any other reasons that led a song to have a peak of success and become a hit. About social influence, the studies use a few features, focusing on a specific social network or a single musical genre. They do not analyze in a more comprehensive way the impact of the social factors on the general scope of musical popularity. Despite the numerous studies conducted, there is still a lot of room for improvement in hit song prediction and, subsequently, there is still much to gain. Hence, there is a strong potential for modeling music success through a broad and suitable combination of the available data. In this sense, our study is innovative because it proposes to merge the music content and subjective attributes of social influence in order to predict success. Furthermore, with recent advances on Big Data and Artificial Intelligence, we believe now it is the time to think big and tackle relevant, hard problems such as this.

3. Methodology

Our motivation is that predicting music success is a complex task, not only due to the numerous features of a song but also due to the social metrics of its artists. Therefore, we plan to deeply explore the most possible features related to the music universe, verify their impact and evaluate different models for predicting popularity. Overall, our proposal follows three tasks: (RI #1) investigation and collection of artists and music-related features; (RI #2 and #3) measurement of the impact and effect of each attribute and their interactions on song's popularity; and (RI #4) development of a method for predicting song popularity.

Investigation and Data Collection To cover the broad availability of music platforms, possible features and information to explore, we plan to collect data from popular platforms, including Billboard, Spotify, Wikipedia and Genius. Therefore, each data source can provide specific and necessary attributes for analyses related to music, artists and albums. Table 1 summarizes the types of information that each of these sources will contribute. The vast majority of data sources will provide valuable resources mainly for the investigation and characterization of attributes related to music and their artists. Regarding the analysis of past and current hit songs, their data will be extracted from rankings available on Billboard charts. Subsequently, some information collected from Spotify, Wikipedia and Genius platforms will be applied to analyze the lyrical and acoustic characteristics of the songs. Finally, from the collected data, we will model music collaboration networks to analyze the collaborative profiles of the artists.

Table 1. Data Sources

Data Source	Description	Information
Billboard	Billboard website provides countless internationally recognized rankings that classify songs and popular albums.	• Billboard charts • Popular songs/albums/artists
Spotify	Spotify streaming platform provides DRM-protected content from record labels and media companies.	• Songs/albums/artists metadata • Acoustic features
Wikipedia	Wikipedia is a free online encyclopedia, created and edited by volunteers around the world.	• Song/artist facts • Trending topics
Genius	Genius is the world's biggest collection of song lyrics and musical knowledge.	• Musical metadata • Song lyrics

Measurement After gathering and characterizing the data collected, we will follow to measure the impact of each attribute on musical popularity. We plan to consider: song content (lyrics) and features, album reputation, artist's reputation and collaboration profiles. To perform such evaluation, we will use statistical techniques and methods to measure the impact of the data. Those techniques allow to identify the role of each attribute in their different levels and to quantify the future impact on the prediction model.

Development After studying, modeling and characterizing the data and measuring the impact of all attributes and their interactions, the ultimate goal is to propose an efficient Machine Learning method to predict if a song will become a hit. For this task, we can evaluate models of regression, classification and similarity, as well as perform the analysis and adjustment of the parameters. Then, we plan to implement the proposed method and validate with popular songs from the existing rankings comparing our method with literature techniques. Finally, the quality of the prediction will be evaluated by using validation techniques (e.g., cross-validation) and execution performance.

4. Current Results

As initial steps, we began to identify collaboration profiles in a musical network composed of successful artists [Silva et al. 2019]. We believe that how artists professionally connect can significantly impact their success. Using data from Billboard and Spotify, we construct a collaborative success-based network to identify distinct profiles, as well as to analyze their impact on artists' popularity. Through topological metrics and clustering algorithms, we identify three well-defined communities with distinct collaboration patterns and notable discrepancies in levels of musical success. We find that successful artists are more likely to have profiles with a high degree of interaction and high diversification. Though the correlation between collaboration profiles and musical success is well established, there has been a lack of research into the causality in this relationship. Therefore, in a current study [Silva and Moro 2019], we assess whether there is a causal relationship between collaboration profiles and artist popularity. Combining the current results with observations from our previous study, we were able to understand more deeply the relationship between artists' collaborative patterns and their musical success.

As future work, we plan to conduct a more accurate analysis on a shorter scale by exploring other metrics for artistic success. Moreover, we are studying other possibilities to deeply investigate techniques of the Pearl Causal Model (PCM). Our preliminary planning for further publications involves the following work and venue: "MusicOSet: An Enhanced Open Dataset for Music Data Mining", *SBBD-Dataset Showcase Workshop*.

References

Araujo, C. V. et al. (2017). Predicting music success based on users' comments on online social networks. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*, pages 149–156, Gramado, RS, Brazil.

Bischoff, K. et al. (2009). Social knowledge-driven music hit prediction. In Huang, R., Yang, Q., Pei, J., Gama, J., Meng, X., and Li, X., editors, *Advanced Data Mining and Applications*, pages 43–54, Berlin, Heidelberg.

Budner, P. and Grahl, J. (2016). Collaboration networks in the music industry. *CoRR*, abs/1611.00377.

Calefato, F. et al. (2018). Collaboration success factors in an online music community. In *Proceedings of the 2018 ACM Conference on Supporting Groupwork*, pages 61–70, Sanibel Island, Florida, USA.

Chon, S. H. et al. (2006). Predicting success from music sales data: a statistical and adaptive approach. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 83–88. ACM.

Dhanaraj, R. and Logan, B. (2005). Automatic prediction of hit songs. In *ISMIR*, pages 488–491.

Iaffaldano, G. (2018). Investigating collaboration within online communities: Software development vs. artistic creation. In *Proceedings of the 2018 ACM Conference on Supporting Groupwork*, pages 384–387.

Kim, Y. et al. (2014). # nowplaying the future billboard: mining music listening behaviors of twitter users for hit song prediction. In *Proceedings of the first international workshop on Social media retrieval and analysis*, pages 51–56. ACM.

Lee, J. and Lee, J.-S. (2015). Predicting music popularity patterns based on musical complexity and early stage popularity. In *Proceedings of the Third Edition Workshop on Speech, Language & Audio in Multimedia*, pages 3–6. ACM.

Pachet, F. and Sony, C. (2012). Hit song science. *Music data mining*, pages 305–326.

Ren, J. et al. (2016). What makes a music track popular in online social networks? In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 95–96. International World Wide Web Conferences Steering Committee.

Salganik, M. J. et al. (2006). Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856.

Silva, M. O. et al. (2019). Collaboration Profiles and Their Impact on Musical Success. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 2070–2077, Limassol, Cyprus.

Silva, M. O. and Moro, M. M. (2019). Causality Analysis Between Collaboration Profiles and Musical Success. In *Proceedings of the 25th Brazilian Symposium on Multimedia and the Web*. [to appear].

Singhi, A. and Brown, D. G. (2015). Can song lyrics predict hits. In *Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research*, pages 457–471.

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Thesis and Dissertations Contest

PROCEEDINGS

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

Program Chair

Caetano Traina Júnior

Editorial

The second Brazilian Theses and Dissertations Contest in Databases (CTDBD) aims at dissemi- nating and award the best doctoral theses and master's dissertations in the database field, approved between January 1st, 2017 and December 31st, 2018 in a Brazilian University.

The competition process consisted of two phases. First, the evaluation committee, composed of experts in the field, elected the two best doctoral theses and the four best master's dissertations, according to their scientific and technological contributions, as well as according to their potential impact on the society and on the state of the art in the Database area. Every submitted manuscript received at least three reviews from selected members of CTDBD's evaluation committee. Great responsibility fell to the preliminary evaluation committee, as they had to choose the theses and dissertations that would be given the opportunity to compete during the second competition round. The recommendations were based on both the full thesis or dissertation and on an extended abstract, summarizing the works' results. It is the set of extended abstracts of the theses and dissertations elected for the competition in the second round that composes this annals. I deeply appreciate all the members of the evaluation committee for the hard work done.

During the second phase, held during SBBD 2019 in Fortaleza-CE, the students will present their work and answer questions from the committee members and audience. From the works selected in the first phase, the committee will chose the best doctoral thesis and the two best master's dissertations. The high quality and diversity of the submitted works made the selection process both highly challen- ging and rewarding. They are a portrait of the high-quality research in the database area developed in our graduate programs.

I would like to thank the committee members for their dedication in providing high quality reviews for the submitted papers. I am also very grateful to all the students and their advisors for the willingness to submit their works to CTDBD. Finally, we thank the local organization committee and the symposium chairs who worked hard to guarantee outstanding discussions. Finally, I wish an excellent event to the whole SBBD community.

Caetano Traina Júnior (Universidade de São Paulo - São Carlos)
Thesis and Dissertation Contest Chair

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

SBBD Steering Commitee

Ângelo Brayner (UFC)
Bernadette Lóscio (UFPE) coordenadora da CEBD
Carina Dorneles (UFSC)
Sérgio Lifschitz (PUC-Rio)
Fábio Porto (LNCC)
Carmem Hara (UFPR)

SBBD 2019 Commitee

Steering Committee Chair
Bernadette Lóscio (UFPE)

Local Chair:
José Maria da Silva Monteiro Filho (UFC, Brazil)

Full Paper Chair
Carina F. Dorneles (UFSC, Brazil)

Short Paper Chair
Fábio Porto (LNCC, Brazil)

Demos and Applications Chair
Robson L. F. Cordeiro (ICMC-USP, Brazil)

Thesis and Dissertation Workshop Chair
Jonice Oliveira (UFRJ, Brazil)

Tutorials Chair
Altigran Soares da Silva (UFAM, Brazil)

Short course Chair
Maria Cláudia Cavalcanti (IME, Brazil)

Workshop Chair
José Antônio Macedo (UFC, Brazil)

Thesis and Dissertation Chair
Caetano Traina Jr. (USP, Brazil)

Graduation Student Workshop Chair
Ticiana Linhares (UFC, Brazil)

Local Organization Committee
SBBD Local Chair: José Maria da Silva Monteiro Filho (DC/UFC)
Leonardo Oliveira Moreira (Instituto UFC Virtual/UFC)
Marum Simão Filho (UNI7)
Angelo Roncalli de Alencar Brayner (DC/UFC)
Javam de Castro Machado (DC/UFC)

Thesis and Dissertation Contest Program Committee
Agma Traina (ICMC-USP)
Altigran Soares da Silva (UFAM)
Angelo Brayner (UFC)
Carina F. Dorneles (UFSC)
Cristina Ciferri (ICMC-USP)
Daniel Kaster (UEL)
Fabio Porto (LNCC)
Javam Machado (UFC)
José Palazzo Moreira de Oliveira (UFRGS)
Maria Camila Nardini Barioni (UFU)
Mirella Moro (UFMG)
Renato Fileto (UFSC)
Ronaldo Mello (UFSC)
Sergio Lifschitz (PUC-Rio)
Valéria C. Times (UFPE)
Vanessa Braganholo (UFF)

Table of Contents (Thesis and Dissertation Contest)

Evolutionary Risk-Sensitive Feature Selection for Learning to Rank

Daniel Xavier de Sousa (author)[1,2], **Thierson Couto Rosa (co-advisor)**[3], **Marcos André Gonçalves(advisor)**[2]

[1]Instituto Federal de Goiás (IFG)
Anápolis – GO – Brazil

[2]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brazil

[3]Instituto de Informática – Universidade Federal de Goiás (UFG)
Goiânia – GO – Brazil

daniel.sousa@ifg.edu.br,mgoncalv@dcc.ufmg.br, thierson@inf.ufg.br

1. Introduction

Learning to Rank (L2R) has established itself as an important research area in Information Retrieval (IR). The L2R task is a central one in many important IR applications such as modern Web search engines, recommendation and question-answering systems [Chapelle et al. 2011]. It applies machine learning algorithms to improve the ranking quality by using annotated information about the relevance of documents.

To maximize results, L2R strategies usually rely on dense representations exploiting dozens of features, some of which are expensive to generate. In several scenarios, some of these features may introduce noise or may be redundant, increasing the cost of the learning process without bringing benefits or even harming the learned ranking model.

Thus, Feature Selection (FS) techniques have been examined in the L2R scenario to improve processing time and increase effectiveness by removing noisy and redundant features. FS indeed may have a high positive impact on processing time in L2R [Chapelle et al. 2011]. In addition to the training time, there is also the cost of constructing the features (actually meta-features) as they are generated by several algorithms (e.g., BM25, PageRank) and some of them need to be computed at query time.

Nevertheless, effectiveness and cost (better summarized by the number of exploited features) are not the only objectives one may want to optimize in a L2R task. In fact, recently the **risk** of getting very poor effectiveness for a few queries with a learned model has gained much attention [Dinçer et al. 2016]. This interest in diminishing risk is due mainly to the fact that users tend to remember the few failures of a search engine very well rather than the many successful searches. In fact, the authors in [Zhang et al. 2014] clearly show that improvements in ranking performance do not always correlate with risk reduction. This has motivated research in *risk-sensitive L2R* which considers the risk aspect of L2R models. The goal of the risk-sensitive L2R task is to enhance the overall effectiveness of a ranking system while reducing the risk of performing poorer than a baseline ranking system for any given query.

Accordingly, in this doctoral dissertation we claim that feature selection used with the intent specifically to enhance efficiency and effectiveness may be a problem to risk-

sensitiveness in L2R. This happens because FS reduces the feature space when considering only overall effectiveness or cost as objectives. It is possible that the reduction of features may worsen the ranking of documents for a few queries (but important ones, such as medical searches), despite improving the ranking for many others. Thus, there may be features that, despite not significantly improving the ranking effectiveness average, enhance the quality of few queries, providing a more robust performance.

Figure 1 provides evidence of the above claim. It shows in the x-axis different rankings (one for each column) when using effectiveness and risk-sensitive measures to sort the features of the MLSR-WEB10K dataset[1]. The first ranking sorts the features considering effectiveness, measured in terms of NDCG@10. The other four rankings correspond to the same features using four different weights of the GeoRisk risk-sensitive function[2]. Each feature corresponds to a colored line in the figure, guided by the rank position of features (in y-axis) in each ranking.

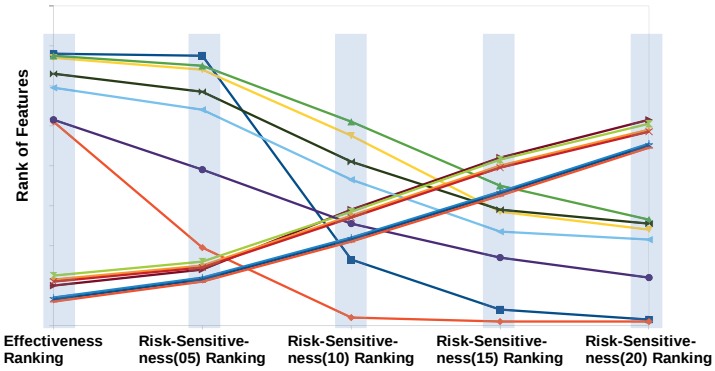


Figura 1. Ranking features when varying the measures of sorting.

Figure 1 shows that some features have an essential behavior on ranking effectiveness, but they are less important from a risk-sensitive perspective, whereas the opposite occurs with other features. In other words, Figure 1 illustrates an essential aspect of FS in L2R: the filtering of features considering only the optimization of effectiveness as a criterion may prune important features that would help to generate more robust (less risky) models. Hence, the problem proposed in this dissertation concerns the selection of features with risk-sensitiveness as a main objective criterion, without loss of effectiveness. Furthermore, as described in our dissertation, the selection of features when using effectiveness as a single objective criterion may incur in higher risk, mainly because the methods tend to optimize an average metric such as Mean Average Precision (MAP) or NDCG, despite potential losses in few points.

1.1. Research Goals

The above observations have motivated us to address distinct objective criteria in FS for the L2R task. To the best of our knowledge, there are no studies that provide a thorough analysis of the impact of feature selection in both effectiveness and risk-sensitiveness in

[1]MLSR-WEB10K is a public dataset, released by Microsoft with 10,000 queries and 136 features.

[2]GeoRisk provides a risk-sensitive evaluation of model performance, by comparing against a set of baselines. The weights ponder the degradation effect or negative variation of the evaluated model against a set of baselines.

the L2R literature. Accordingly, the novel proposed objective criteria are: i) *maximizing* the ranking effectiveness; ii) *minimizing* the risk for most queries; and iii) *reducing* the feature space dimensionality, *all at the same time*. Moreover, we analyze the impact of FS for L2R in these objectives considering them both individually (as single objectives), as well as combined as multi-objectives to be optimized.

By considering a robust and effective evaluation with FS, our dissertation aims to obtain a possibly smaller set of features that guarantees ranking effective and risk-sensitive performance. This is in contrast to existing FS for L2R approaches whose goal is to drastically reduce the number of features in order to control the processing time.

We also propose a novel methodology to assess the impact on effectiveness and risk-sensitiveness when diverse (most of the times, conflicting) objective criteria are applied to the FS for the L2R task. Using an efficient and effective wrapper strategy, our proposed methodology explores diverse sets of features as a search space and uses an evolutionary search to select the best features set according to single or multi-objective criteria. Wrapper strategies are traditionally recognized as time consuming approaches. To deal with this issue, in our dissertation we propose to exploit "cheap" weak-learners as black-boxes that make the process more scalable and less costly. As a positive side effect, weak-learners also promote diversity in the solutions. In other words, we drive the exploration of the space of solutions using improvements in both wrapper and multi-objective optimization processing.

In summary, in our dissertation we provide four novel contributions:

1. We open up a new perspective of FS for L2R, which highlights the importance of considering risk as an explicit objective criterion. In this context, we are not only considering the average effectiveness obtained by a drastically reduced subset of features, but a subset that provides a risk-sensitive and effective performance;
2. We introduce single and multi-objective criteria to perform FS for L2R, considering three important objectives, *concomitantly*: feature dimensionality reduction, effectiveness and risk-sensitiveness. Some of these (conflicting) objective criteria were never evaluated in FS for L2R;
3. A novel efficient and effective evolutionary methodology to evaluate different objective criteria in FS for L2R. We apply weak-learners (apparently counter intuitive) to decrease the execution time while increasing diversity, and a paired test comparison over a multi-objective search to provide an accurate set of features.
4. We provide a broad discussion of the proposed methodology and objectives, showing that, in FS for L2R, distinct goals (with feature reduction or accuracy) can be achieved by varying the objective criteria. Also, most previous works explored only small datasets, and we consider large ones, e.g. WEB10K and YAHOO.

During the dissertation, we published some papers in the world leading Information Retrieval conferences and journals, such as [Sousa et al. 2016]*(A1)* and [Sousa et al. 2019]*(A2)*. Beside them, we also published other papers in L2R area, as [Sousa et al. 2012]*(B1)*, [Freitas et al. 2016]*(B3)*[3], and [Freitas et al. 2018]*(A2)*.

[3]Selected as the best paper of the conference

2. A Novel Feature Selection Methodology

This dissertation proposes a new FS methodology for L2R, with regard to both: the objective criteria and the execution process. Differently from all other works in the literature of FS for L2R, we here describe the performance of many objective criteria from a risk-sensitive perspective and we show that risk-sensitiveness is an important objective criterion in FS. We consider many objective criteria over three dimensions, *concomitantly*: ranking performance, dimensionality and risk-sensitiveness.

In order to optimize theses objective criteria, our methodology explores a wrapper strategy in a Evolutionary Search. In fact, Evolutionary Algorithms (EAs) are well suited to estimate the impact of the distinct proposed objective criteria and also to evaluate our statements, mainly due to their capability of obtaining non-linear ranking functions. More specifically, to investigate combinations of simultaneous objectives, our proposal uses a multi-objective criteria approach based on Pareto frontier optimization. There are several general-purpose multi-objective optimization methods that can be used in this case. We have chosen Strength Pareto Evolutionary Algorithm (SPEA2) [Zitzler et al. 2001], which besides being the state-of-the-art in multi-objective optimization, has already been successfully applied to several related problems [Li et al. 2015].

However, in many works the computation of the fitness value for an individual in a wrapper strategy is time consuming, as it is necessary to construct a hard L2R model with a subset of features corresponding to the individual and to evaluate the model, specially for some large datasets and state-of-the-art L2R algorithms. One of the key points in our work is the reduction of the searching time during the wrapper-based feature selection, by applying a weak-learner to optimize a cost-function that evaluates individuals over the evolutionary search. Basically, the intuition of weak-learner quality in evolutionary search regards to its capability to evaluate the importance of bad features among the individuals during the evolutionary process and not only to find the individual that obtain the maximum performance. In contrast, strong-learners can decrease the weight of bad features while build the model, producing similar effectiveness when comparing distinct sets of features.

In addition, the literature shows that the Pareto frontier set can be large, especially when two objectives are conflicting. This can make the selection within the Pareto set very hard, decreasing the final performance. We address this selection using a strict comparison over the individuals by means of an evolutionary search, using statistical hypothesis tests. As a result, our method provides a smaller Pareto set with only statistically superior individuals.

To summarize, we propose a novel (original) FS methodology for LR2, combining new objective criteria, the wrapper strategy with weak-learners and the strict comparison by means of paired statistical tests.

3. Experimental Evaluation: Summary

As our work explores new directions in FS for L2R, we performed an extensive series of experiments to validate our hypotheses and claims. One of our main results show that the tested objective-criteria can indeed improve risk-sensitiveness without decreasing the effectiveness and reducing the feature space. Table 1 summarizes the results in terms

	Effectiveness	Risk-sensitiveness	Feature Reduction
$\overset{\text{E-R}}{\succ}$	73	75	3
$\overset{\text{E-G}}{\succ}$	62	69	13
$\overset{\text{G}}{\succ}$	62	63	8
$\overset{\text{E}}{\succ}$	54	56	27
$\overset{\text{E-F-G}}{\succ}$	49	54	41
$\overset{\text{T}}{\succ}$	46	53	32
$\overset{\text{G-F}}{\succ}$	45	50	42
DivFS	27	35	55
BTFS	21	24	84
$\overset{\text{T-F}}{\succ}$	10	9	71
$\overset{\text{E-F-R}}{\succ}$	7	10	68
$\overset{\text{E-F}}{\succ}$	4	7	74

Tabela 1. ***Heatmap of proposed objective criteria for FS in L2R.***

of effectiveness (E), risk-sensitiveness (R, T and G, respectively for URisk, TRisk and GeoRisk measures [Dinçer et al. 2016]), and feature reduction (F) when varying the objective criteria. The Table shows the number of (statistically significant) victories for each objective-criterion compared to all others, varying the datasets (WEB10K, YAHOO, TD2003 and TD2004) and the weak learners (Linear Regression and Regression Tree as black-boxes). In the Table, higher numbers correspond to darker colors and more victories[4]. Overall, the objective criteria that combine effectiveness and risk-sensitiveness (e.g., $\overset{\text{E-R}}{\succ}$ and $\overset{\text{E-G}}{\succ}$) achieve a higher number of victories in terms of risk-sensitiveness, with additional gains in effectiveness and minor feature reduction. On the other hand, effectiveness as sole objective-criterion, ($\overset{\text{E}}{\succ}$), produced a larger feature reduction but with losses in terms of effectiveness and risk-sensitiveness.

In Table 1, we can also observe that methods that applied a more drastically feature reduction, or methods that exploit the reduction of the number of features as an explicit objective ($\overset{\text{T-F}}{\succ}$, $\overset{\text{E-F-R}}{\succ}$ and $\overset{\text{E-F}}{\succ}$) could not achieve significant improvements in effectiveness and risk-sensitiveness ($\overset{\text{E-R}}{\succ}$ and $\overset{\text{E-G}}{\succ}$). Finally, methods focused on how to provide a more accurate model (effectiveness) and low-risk (e.g. $\overset{\text{E-R}}{\succ}$, $\overset{\text{E-G}}{\succ}$), could obtain a very interesting balance between effective and robust performance, with additional gains in feature set reduction. Note that our methods have a much broader goal than alternatives in the literature (e.g. DivsFS[Naini and Altingovde 2014] and BTFS[Pan et al. 2011]), whose focus in only on reducing the number of features for the sake of controlling noise, redundancy and processing time.

4. Conclusion

Ours is the first dissertation that thoroughly investigated the impact of risk-sensitiveness in feature selection for Learning to Rank. In this context, we proposed several relevant contributions. We perform a multi-objective criteria using SPEA2 as a general multiobjective

[4]The maximum number of victories is 88: 11 strategies with 4 datasets and 2 weak-learners. There may be statistical ties between the strategies.

criteria, concerning the interaction of features on a wrapper strategy. We noted that this strategy provides flexibility to search for several regions in the feature space, being able to achieve feature reduction without exploiting number of features as an explicit objective criterion. Our methodology extends the evolutionary wrapper algorithms by using weak learners as black-boxes. We show that weak learners can be applied in a wrapper strategy to perform FS on the L2R task with speedups of more than 120x, without decreasing effectiveness[5]. Our methodology also extends works in exploiting the Pareto set, as it applies strict comparison among individuals, by performing paired statistical tests to define the dominance relationship of the individual over the generations. As described in our experiments, this strategy reduces the conflict between individuals in multi-objective criteria, decreasing the Pareto set dimension over 50% while improving the effectiveness of the selected individual. In our dissertation we stress the evaluation of several risk-sensitive measures with effectiveness and number of features as multi-objective to perform feature selection. As a result, we show that using effectiveness and risk-sensitiveness as objective criteria produces a better subset of features for L2R, demonstrating the importance of risk-sensitiveness as an explicit objective criterion in FS for L2R.

As main future work, we envision the application of our novel methodologies and strategies in any problem in which maximising effectiveness (by means of L2R strategies) and minimizing cost (number of feature) and risk are important. Recommendation, question answering, multimedia retrieval, classification, are just a few of numerous possibilities. Other possible extension is the consideration of other objectives (e.g., novelty, diversity) other than the ones mentioned above.

Referências

Chapelle, O., Yi, C., and Liu, T.-Y. (2011). Future directions in learning to rank. *In YLRC*, pages 129–136.

Dinçer, B. T., Macdonald, C., and Ounis, I. (2016). Risk-Sensitive Evaluation and Learning to Rank using Multiple Baselines. *In SIGIR*, pages 483–492.

Freitas, M., Sousa, D., Martins, W., Couto, T., Silva, R., and Gonçalves, M. (2016). A Fast and Scalable Manycore Implementation for an On-Demand Learning to Rank Method. *In WSCAD*, 1:1–12.

Freitas, M., Sousa, D., Martins, W., Couto, T., Silva, R., and Gonçalves, M. (2018). Parallel rule-based selective sampling and on-demand learning to rank. *In CCPE*, pages 1–12.

Li, B., Li, J., and Tang, K. (2015). Many-Objective Evolutionary Algorithms: A Survey. *In CSUR*, 48:1–35.

Naini, K. D. and Altingovde, I. S. (2014). Exploiting Result Diversification Methods for Feature Selection in Learning to Rank. *Proceeding of the 36th European Conference on Information Retrieval - ECIR*, pages 455–461.

Pan, F., Converse, T., Ahn, D., Salvetti, F., and Donato, G. (2011). Greedy and randomized feature selection for web search ranking. *CIT*, pages 436–442.

Sousa, D., Canuto, S., Couto, T., Martins, W., and Gonçalves, M. (2016). Incorporating Risk-Sensitiveness into Feature Selection for Learning to Rank. *In CIKM*, pages 257–266.

Sousa, D., Canuto, S., Gonçalves, M. A., Couto, T., and Martins, W. (2019). Risk-sensitive learning to rank with evolutionary multi-objective feature selection. *In TOIS*, 37:24:1–24:34.

Sousa, D., Couto, T., Martins, W., Silva, R., and Gonçalves, M. (2012). Improving on-demand learning to rank through parallelism. *In WISE*, pages 526–537.

Zhang, P., Hao, L., Song, D., Wang, J., Hou, Y., and Hu, B. (2014). Generalized Bias-Variance Evaluation of TREC Participated Systems. *In CIKM*, pages 3–6.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. *In EUROGEN*, pages 12–19.

[5]After a set of features is selected, a state-of-the-art L2R algorithm is used as the final model.

Runtime Dataflow Analysis in Scientific Applications

D.Sc.: Vítor Silva[1*] at PESC/COPPE/UFRJ (June 2018)

Supervisor: Marta Mattoso[1]
Co-supervisors: Daniel de Oliveira[2], Patrick Valduriez[3]

[1]Universidade Federal do Rio de Janeiro, Brazil (PESC/COPPE/UFRJ)

[2]Universidade Federal Fluminense, Brazil (IC/UFF)

[3]INRIA and LIRMM, Montpellier, France

marta@cos.ufrj.br, vitor.sousa@dell.com, danielcmo@ic.uff.br
Patrick.Valduriez@inria.fr

Abstract. *Much of the data produced by programs of scientific applications need to be analyzed by scientific domain users to validate their hypotheses. However, it is not trivial since other ad-hoc programs must be developed to access and to capture these scientific data. In most cases, users need to relate raw data from different simulation programs for runtime analysis. This thesis proposes a dataflow abstraction that represents, monitors, debugs, and analyzes the data element flow produced by different simulation programs. This abstraction has an algebra implemented as lightweight dataflow monitoring components to be invoked by high performance applications. In several experiments with real data, it provided complex runtime queries helping users to fine-tune parameters at runtime, all with a negligible overhead.*

1. Introduction

Scientific or Computational Science and Engineering (CSE) applications are based on computational models that solve problems typically requiring High Performance Computing (HPC) [Rüde et al. 2016]. CSE applications are not tied to a particular domain. They can be found in biology, chemistry, geology, several engineering areas, etc. They have the exploratory nature of scientific applications and also have to deal with large-scale executions, which last for a long time even when using HPC. The software ecosystem for developing these applications involves much more than writing scripts or invoking a chain of legacy scientific codes. Computational scientists develop their simulation codes based on complex mathematical modeling that results in invoking components of CSE frameworks and libraries. For example, components are invoked to provide for: (i) support for partial differential equation discretization methods like libMesh, FEniCS, MOOSE, deal.II, GREENS, OpenFOAM; (ii) algorithms for solving numerical problems with parallel computations, like PETSc, LAPACK, SLEPc; (iii) runtime visualizations, like ParaView Catalyst, VisIt, SENSEI; (iv) parallel graph partitioning, like ParMetis, Scotch; and (v) I/O data management like ADIOS.

[*] Vitor Silva Sousa is currently at Dell-EMC Research Brazil

As a result, a typical CSE software requires invoking functions, components, or APIs from these libraries or frameworks. Several parameters have to be set to invoke these highly efficient components, which are very difficult to preset and need monitoring with data analysis for fine-tuning. Parameter setting is a complex problem also in non-CSE applications, like hyperparameters in Deep Learning applications. It involves analyzing application data with several relationships, which are hard to define at runtime.

By observing a specific pattern, an experienced interpreter can infer that something is not going well in the simulation, deciding to stop it or change parameters, preferably at runtime, resuming or adapting the simulation. However, to do that, the visualization should be complemented with information regarding the evolution of Quantities of Interest (QoI), such as residual norms, number of linear and nonlinear iterations, number of epochs, often within a specific time window, not just the current values. To obtain this complementary information, even the experienced interpreter has difficulty in identifying the files related to the time window, opening and parsing files to obtain specific values, filtering, aggregating and tracking their evolution. For example, a typical scenario generates "isolated" data in raw data files, visualization files, provenance databases and monitoring data in log files. All these data have implicit relationships, which makes both runtime and post-processing analyses very complex.

This thesis proposes an approach that monitors, debugs, and analyzes the data element flow produced by different simulation programs. It presents a dataflow-based algebra implemented as a component-based architecture, named as ARMFUL [Silva et al. 2016a, Silva et al. 2017], to extract, index and relate scientific data produced in these several simulation steps. The standard W3C PROV is adopted to represent the dataflow of scientific applications. Provenance data is a natural way of representing the data derivation path of the data produced by the applications. As a result, the application becomes provenance-aware. The resulting provenance database is an important asset in data quality, trust and reproducibility, but this thesis goes one step further and provides provenance databases with rich domain data ready to be queried at runtime.

Despite the several solutions available for making applications provenance-aware [Stamatogiannakis et al. 2016, Moreau et al. 2018, Pimentel et al. 2017], capturing provenance data in CSE applications is still an open, yet important, issue. The challenges are mainly related to performance and provenance granularity. [Stamatogiannakis et al. 2016] evaluated tradeoffs in provenance capture mechanisms. They consider that solutions that are easy to deploy, collect provenance in a very fine grain and present a significant overhead, while solutions that are based on function calls present low overhead and granularity is controlled by a code adaptation. The disadvantage of inserting function calls is the need to have access to the code. This is not an issue in CSE applications as very often the code to be adapted is a script written by the computational scientist, who can assist in inserting the calls, as already is done with specialized libraries.

The main contributions of this thesis are a dataflow abstraction and techniques for scientific data modeling, capture, indexing and analyses. ARMFUL's components can be instantiated on a scientific workflow system (*e.g.,* A-Chiron) or a library of components (*e.g.*, DfAnalyzer). We evaluate these instances using real CSE applications from different application domains like Astronomy, Bioinformatics, Oil&Gas, and also business, all in HPC environments. The experimental results evidenced a negligible overhead added to the scientific application execution time in all domains. The resulting

dataflow has been analyzed by real users in several runtime queries, which allowed them to fine tune their executions improving the overall lifecycle of the application.

2. Runtime Data Analysis

The thesis defines a dataflow abstraction for representing relationships between datasets manipulated by computational models. The smallest unit of interest is the *data element* (e). A data element has values (v) for each predefined attribute (a) that represents e. The schema that represents e is a set A, where each a is represented as (name, type). A set of data elements consists of a *data collection* (c). Then, a *dataset* (s) is composed of a set of data collections (C). A *data transformation* (t) consumes data from one (or more) dataset(s) as input (S_{input}) and produces data in one (or more) dataset(s) as output (S_{output}). Furthermore, two data transformations can present a *data dependency* (φ) with relation to a dataset, when the data is produced by one data transformation ($t_{previous}$) and consumed by another (t_{next}). Based on such concepts, a *dataflow* (D_F) is defined by the data resulting from the composition of data transformations (T), manipulating datasets (S) concerning data dependencies (Φ). These dataflow abstractions are represented in PROV-Df [Silva et al 2016, Silva et al. 2017], a W3C PROV-compliant data model, which is agnostic concerning the scientific application domain.

The thesis innovates by extracting and relating raw data (*e.g.*, QoI) from heterogeneous distributed files, at runtime. A-Chiron [Silva et al. 2017] and DfAnalyzer[1] [Silva et al. 2018] access and extract strategic domain data associated to these files using in situ and in transit raw data extraction approaches. Once extracted, raw data is related using the dataflow abstraction and follows PROV-Df to store them in a provenance database, which acts as a global map of the CSE application raw data. The provenance database is managed by a columnar DBMS, which can be queried at any moment while the application executes. This dataflow map allows for monitoring queries like *what is the average error estimate calculated in all iterations so far* or *what are the areas of interest in a mesh that should have larger or smaller time steps*. This helps in fine-tuning tolerances and other user steering actions, as described in real DfAnalyzer use cases of scripts built with libMesh library [Camata et al. 2018, Silva et al. 2016a], with FEnICS [Silva et al. 2016b, Alnæs et al. 2015] and a toy business application [Silva et al. 2018a] using Spark. The same kind of data analysis was performed in a real astronomy workflow modeled and executed using A-Chiron [Silva et al. 2016a].

A-Chiron follows our dataflow abstraction to extend the Chiron parallel scientific workflow system for providing raw data extraction and indexing with two algebraic operators and using *ad-hoc* programs or third-party tools, such as FastBit and RAW. DfAnalyzer is a component-based reference architecture for dataflow analysis. DfAnalyzer has six components: (i) Provenance Data Extractor (PDE); (ii) Raw Data Extractor (RDE); (iii) Raw Data Indexer (RDI); (iv) Dataflow Viewer (DfViewer); (iv) Query Interface (QI); and (vi) Database (DfDB). DfAnalyzer captures provenance and domain-specific data (*i.e.*, strategic data obtained during the application execution). DfAnalyzer enables raw data extraction from such files and content indexing by direct access to memory or invoking third-party programs or tools. The first three components are invoked by plugging calls on the application, while the other two have independent

[1] https://hpcdb.github.io/armful/dfanalyzer.html

interfaces for the user to submit data analyses at runtime. Therefore, A-Chiron is recommended when scientific domain users have applications that need to be parallelized as well as raw data extraction to improve data analysis, while DfAnalyzer is recommended for CSE scripts that are already parallel and use HPC libraries.

3. Impact and Conclusions

Encapsulating all data analytics support in one specific library/component makes the application autonomous and data analyses can always be switched off. There are several advantages of this dataflow analysis approach. First, simulation data is preserved in their format and not fully replicated in the DBMS. Second, data is related among different files while it is being generated, which might be cumbersome after the simulation ends, as in post-processing approaches. Third, the history of data generation is registered for further analysis or reproduction through provenance, following W3C PROV. Fourth, efficient data management techniques from relational DBMS (*i.e.,* PostgreSQL and MonetDB) can be used at runtime. Consequently, this thesis proposes a domain independent dataflow approach, which provides a provenance database enriched with quantities of interest (*i.e.,* domain data) that can be queried at runtime with negligible overhead.

This thesis contributes to improve developing software for CSE applications because the developer does not have to write code for each data analysis and for logging data. The same data model can be reused, and the databases associated to raw data from all the executions can be further analyzed using AI tools to improve CSE parameter settings. A-Chiron and DfAnalyzer provide data for designing recommendation systems and finding correlations among distributed data. To improve the usability and reproducibility of the dataflow analysis approach, DfAnalyzer is available for download with documentation, tutorials, examples and a container to help its deployment.

We validated our solution in a real sedimentation simulation in the Oil and Gas domain. In this simulation, we exposed the relationships among data transformations with complex mathematical models. We could also see that data analysis at runtime allowed for the parameter-tunings, which not only reduced the execution time by 10 days (37%), but also made it finish successfully. If no adaptation happened, the execution would raise an out-of-memory error and simulation would be interrupted. Therefore, user steering support provided in this thesis contributes to the process of scientific discovery and explainability by automatically adding provenance, context and relationships to scientific data native formats and helping users in their decision-making process. Moreover, all dataflow analyses performed in this thesis are done with provenance data capture techniques with a negligible overhead, always below 1%, while provided reductions of approximately 52% in the overhead of data ingestion with our indexing techniques.

These results originated several publications. Several journal papers [Silva et al. 2016a, Silva et al. 2017, Silva et al. 2018a, Silva et al. 2018b, Camata et al. 2018]. The results also originated several conference papers [Silva et al. 2016b, Silva et al. 2016c, Silva et al. 2018c, Ocaña et al. 2015]. We highlight the papers in important venues such as the International Provenance and Annotation Workshop (IPAW) and Workflows in Support of Large-Scale Science Workshop (WORKS). Vítor also collaborated with several colleagues [de Oliveira et al. 2015, Guedes et al 2017, Guedes et al. 2018a, Guedes et al. 2018b, Guedes et al. 2018c, Marinho et al. 2017, Pina et al. 2017, Souza et al. 2015, Souza et al. 2016, Souza et al. 2017[a], Souza et al. 2017b, Souza et al. 2019].

References

Alnæs, M., Blechta, J., Hake, J., Johansson, A., Kehlet, B., et al. (2015) Archive Of Numerical Software: The FEniCS Project Version 1.5. University Library Heidelberg.

Camata, J.J. ; Silva, J.J. ; Valduriez, P. ; Mattoso, M. ; Coutinho, A.L.G.A. (2018). In situ visualization and data analysis for turbidity currents simulation. Computers & Geosciences. v. 110, pp. 23–31.

de Oliveira, D., Silva, V., Mattoso, M., (2015), "How Much Domain Data Should Be in Provenance Databases?". In: Workshop on Theory and Practice of Provenance (TaPP), Edinburgh, Scotland.

Guedes, T. ; Silva, V ; Camata, J J. ; Mattoso, M ; Oliveira, D . (2017) Análise de Dados Científicos: uma Análise Comparativa de Dados de Simulações Computacionais. In: Simpósio Brasileiro de Banco de Dados, 2017, Uberlândia. 32o SBBD. p. 222-227.

Guedes, T. ; Silva, V. ; Mattoso, Marta ; Oliveira, D. (2018a). Análise Online de Dados de Proveniência e de Domínio de Aplicações Spark com SAMbA. In: XXXIII Simpósio Brasileiro de Banco de Dados, 2018, Rio de Janeiro. SBBD 2018. p. 17-22.

Guedes, T. ; Silva, Vítor ; Camata, José J. ; Bedo, M. V. N. ; Mattoso, Marta ; Oliveira, Daniel . (2018b) Towards an Empirical Evaluation of Scientific Data Indexing and Querying. Journal of Information and Data Management - JIDM, v. 9, p. 84-93.

Guedes, T ; Silva, V ; Mattoso, M ; V. N. Bedo, M ; De Oliveira, D . (2018c) A Practical Roadmap for Provenance Capture and Data Analysis in Spark-Based Scientific Workflows. In: 2018 IEEE/ACM Workflows in Support of LargeScale Science (WORKS), 2018, Dallas. 2018 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS), 2018. p. 31-41.

Lavril, T. ; Mattoso, M ; Silva, V ; Costa, D. ; Rochinha, F ; Miras, T. ; Coutinho, A L.G.A. (2016). Controlling Parallel Adaptive Sparse Grid Stochastic Collocation Simulations with Workflows. In: Iberian Latin American Congress on Computational Methods in Engineering, 2016, Brasilia. XXXVII CILAMCE. p. 1-12.

Marinho, A ; De Oliveira, D ; Ogasawara, E ; Silva, V ; Ocaña, K ; Murta, L ; Braganholo, V ; Mattoso, M. (2017) Deriving scientific workflows from algebraic experiment lines: A practical approach. Future Generation Computer Systems, v. 68, p. 111-127, 2017.

Moreau, L., Batlajery, B.V., Huynh, T.D., Michaelides, D., Packer, H. (2018) A Templating System to Generate Provenance. IEEE Trans. Softw. Eng. 44, 103–121 (2018).

Ocaña, K., De Oliveira, D., Silva, V., et al., (2015), "Data Analytics in Bioinformatics: Data Science in Practice for Genomics Analysis Workflows". In: IEEE International Conference on eScience, Munich, Germany.

Pimentel, J.F., Murta, L., Braganholo, V., Freire, J. (2017) noWorkflow: a tool for collecting, analyzing, and managing provenance from python scripts. Proc. VLDB Endow. 10, 1841–1844 (2017).

Pina, D. ; Campos, V. ; Silva, V ; Ocaña, K. ; Oliveira, D. ; Mattoso, M . (2017) BioSciCumulus: um portal para análise de dados de proveniência em workflows de biologia computacional. In: Brazilian e-Science Workshop, 2017, Sao Paulo. 11º BreSci - Brazilian e-Science Workshop - CSBC 2017, 2017. p. 1096-1103.

Rüde, U., Willcox, K., McInnes, L.C., Sterck, H.D., Biros, G., et al. (2016) Research and Education in Computational Science and Engineering. CoRR. abs/1610.02608, (2016).

Silva, V.; de Oliveira, D.; Valduriez, P.; Mattoso, M. (2016a) Analyzing related raw data files through dataflows. CCPE, v. 28, p. 2528-2545.

Silva, V., Camata, J., De Oliveira, D., et al., (2016b), "In Situ Data Steering on Sedimentation Simulation with Provenance Data". In: Poster session of Supercomputing conference, Salt Lake City, UT, USA.

Silva, V., Neves, L., Souza, R., et al., (2016c), "Integrating domain-data steering with code-profiling tools to debug data-intensive workflows". In: Workshop on Workflows in Support of Large-Scale Science (WORKS), Salt Lake City, Utah, USA.

Silva, V.; Leite, J; Camata, J; de Oliveira, D; Coutinho, A.L.G.A ; Valduriez, P; Mattoso, M. (2017) Raw Data Queries during Data-intensive Parallel Workflow Execution. Special Issue on Workflows for Data-Driven Research in the FGCS Journal. v. 75, pp. 402-422, 2017.

Silva, V. ; De Oliveira, D. ; Valduriez, P. ; Mattoso, M. (2018a) DfAnalyzer: Runtime Dataflow Analysis of Scientific Applications using Provenance. PVLDB Journal, v. 11(12), pp. 2082-2085, 2018a.

Silva, V ; Neves, L ; Souza, R ; Coutinho, A ; De Oliveira, D ; Mattoso, M . (2018b) Adding domain data to code profiling tools to debug workflow parallel execution. Future Generation Computer Systems-The International Journal of eScience.

Silva, V ; Souza, R ; Camata, J J. ; Oliveira, D. ; Valduriez, P ; Coutinho, A. L.G.A. ; Mattoso, M. (2018c) Capturing Provenance for Runtime Data Analysis in Computational Science and Engineering Applications. In: IPAW, 2018, Londres. 7th International Provenance and Annotation Workshop, LNCS. v. 11017. p. 183-187

Souza, R. F. ; Silva, V. ; Neves, L. ; Oliveira, D. ; Mattoso, M . (2015) Monitoramento de Desempenho usando Dados de Proveniência e de Domínio durante a Execução de Aplicações Científicas. In: XIV Workshop em Desempenho de Sistemas Computacionais e de Comunicação, 2015, Recife. WPerformance, CSBC, 2015.

Souza, R. F. ; Silva, V. ; Coutinho, A. L. G. A. ; Valduriez, P ; Mattoso, M . (2016) Online Input Data Reduction in Scientific Workflows. In: 12th Workshop on Workflows in Support of Large-Scale Science, 2016, Salt Lake City. (WORKS 2016) in ACM/IEEE Supercomputing SC'16 Workshops, 2016. v. 1800. p. 44-53.

Souza, R. F. ; Silva, V ; Miranda, P. ; Lima, A A.B. ; Valduriez, P ; Mattoso, M . (2017a) Spark Scalability Analysis in a Scientific Workflow. In: Simpósio Brasileiro de Banco de Dados, 2017, Uberlândia. 32o SBBD. p. 288-293.

Souza, R. F. ; Silva, V. ; Camata, J J. ; Coutinho, A. L. G. A. ; Valduriez, P ; Mattoso, M . (2017b) Data reduction in scientific workflows using provenance monitoring and user steering. Future Generation Computer Systems-The International Journal of eScience, 2017. (published online: http://dx.doi.org/10.1016/j.future.2017.11.028)

Souza, R ; Silva, V ; Camata, J J. ; Coutinho, A L.G.A. ; Valduriez, P ; Mattoso, M . (2019) Keeping track of user steering actions in dynamic workflows. Future Generation Computer Systems-The International Journal of eScience, v. 99, p. 624-643, 2019.

Stamatogiannakis, M., Kazmi, H., Sharif, H., Vermeulen, R., Gehani, A., Bos, H., Groth, P.: Trade-Offs in Automatic Provenance Capture. In: IPAW, pp. 29–41 (2016).

SPST-Index: A Self-Pruning Splay Tree Index for Caching Database Cracking

Pedro Holanda - Advisor: Eduardo Almeida

[1]UFPR
Curitiba - Brazil

{holanda}@cwi.nl

1. Introduction

Database Cracking [3] presents a self-organizing database partitioning for column-oriented relational databases. It works by physically self-organizing database columns into partitions, called cracked pieces. The goal is to create cracked pieces for all accessed intervals of range queries. Cracker indices are created to keep track of these partitions.

In contrast to usual indices in the literature, the nodes of a cracker index do not point to all the disk blocks of a column. Instead, they point to the beginning of each cracked piece to boost access to an interval of values.

The current data structure implemented as cracker index is the self-balancing AVL Tree [1], where the height of the adjacent children subtrees of any node differ by at most one. However, this property makes the AVL tree particularly cache-inefficient. The tree nodes accessed only for a few times (i.e, "Cold Data") and the most accessed ones (i.e, "Hot Data") are spread all over the index. Another concern lies in the index size as "Cold Data" are kept in the index. Eventually, the cracker index converges to a full index (i.e, all values indexed) with high administration costs for high-throughput updates.

In this work, we present a data structure called Self-Pruning Splay Tree (SPST) to index database cracking and keep "Hot Data" close to the root of the tree. The SPST is a Splay Tree variant with a self-adjusting property carried out by the *splaying* operation. *Splaying* consists of a sequence of rotations to move a node way up to the root of the tree. To every range query, our algorithm rotates the nodes pointing to the edges and to the middle value of the predicate interval. With "Hot Data" constantly rotated, they eventually remain close to the root. On the other hand, "Cold Data" are stored close to the leaves presenting the opportunity to prune them out of the index and improve maintenance and update costs.

This work was published as a short paper at EDBT - 2017[1]. We provide Open-Source implementations of each of the techniques we describe and their benchmarks.[2].

2. Related Work

Database Cracking [4] (also known as "Adaptive Indexing") is the original adaptive indexing technique. It works by physically reordering the index while processing queries. It consists of two data structures: a cracker column and a cracker index. Each incoming query cracks the column into smaller pieces and then updates the cracker index with the

[1]https://openproceedings.org/2017/conf/edbt/paper-285.pdf
[2]Our implementations and benchmarks are available at https://github.com/pholanda/SPST

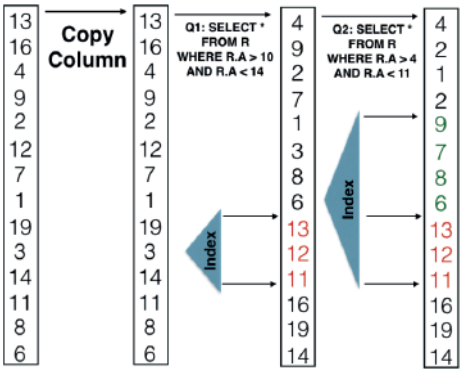
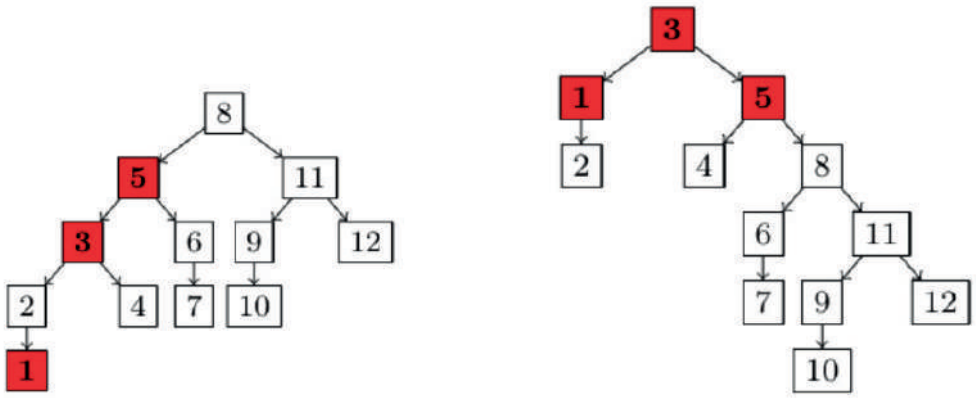


Figure 1. Database Cracking

reference to those pieces. As more queries are processed the cracker index converges towards a full index. This process is visualized in Figure 1.

There are many data structures in the literature to keep track of data partitions. In database cracking the AVL is the data structure of choice, but other self-balancing trees, like RedBlack or 2-3 trees, draw the same result. These trees have the property of keeping the height of the tree for self-balancing purposes. However, this property makes them cache-inefficient for range queries. The tree nodes accessed only for a few times and the most accessed ones are spread all over the tree.



(a) Original Tree (b) After Lookup

Figure 2. SPST Range Splay

3. The SPST-Index

Our contribution regards recognizing "hot data" to improve data access and recognizing "cold data" to prune unused data and boost updates.

3.1. Splaying

A Splay Tree [5] is a self-adjusting binary search tree that uses a splaying technique every time a node is searched, updated, inserted or deleted. *Splaying* consists of a sequence of rotations that moves a node to the root of the tree. Lookup, insertion, and deletion take

$O(\log n)$ time in the average and worst case scenarios, where n is the number of nodes in the Splay Tree. It clusters the most accessed nodes near the root of the tree. Therefore, the most frequent accessed nodes will be accessed faster. Since we are dealing with range queries, our goal is to splay the query range, instead of splaying only one node like the original splay tree. The self-adjustment algorithm in our data structure is straightforward: we first splay the leftmost node of the range, then the rightmost node and later the closest node to the middle.

Let us consider for cracker index the SPST depicted by Figure 2. If a range query of $1 < A < 5$ is executed, the algorithm performs three operations: Splay (1), Splay (5) and Splay ($\lceil \frac{(1+5)}{2} \rceil$). Figure 2(b) depicts the resulting tree with nodes 1, 3 and 5 close to the root. In the SPST, the nodes remain close to the root as long as they are frequently accessed. In our index, the nodes pointing to the most accessed cracked pieces remain close to the root.

(a) Original Tree (b) After Prune

Figure 3. SPST Prune

3.2. Pruning

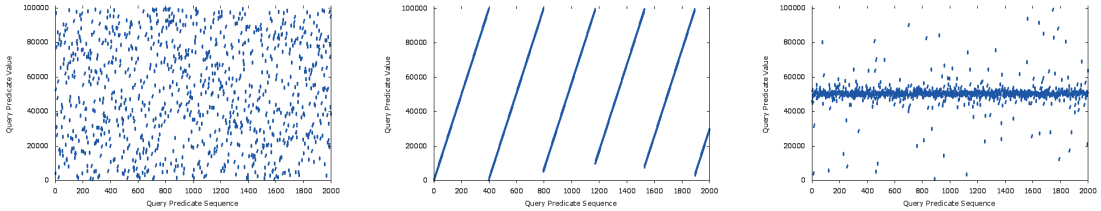
Besides speeding up the access to hot data, another goal is to speed up updates and maintenance costs when rotating hot data. We assume that eventually the nodes stored at the leaves point to cold data. The maintenance strategy of our data structure is to prune the leaves. As we prune them, the update time is expected to shrink. The downside of pruning the tree is that the following queries can become slightly more expensive compared to the situation where we do not have any pruning at all. Our hypothesis is that we mitigate this cost with the gains in the update time. When we prune the leaves, the size of the index shrinks, in the best case, to $\lfloor \frac{n}{2} \rfloor$, where n is the number of nodes in the SPST.

Let us suppose the SPST index depicted by Figure 3(a). In this scenario the most frequent range is between 10 and 30. Let us suppose inserting the value 21 in the Cracker Column. To do this, we need to update the nodes 35, 30 and 25 respective pointers to the cracker column and merge at their respective cracker column pieces. Instead, we start pruning the leaves having as result the tree depicted by Figure 3(b). Then we only need to update the pointer to the cracked piece of node 30.

4. Experimental Analysis

In this section, we discuss the results of our experimental evaluation of the SPST implemented as a cracker index. We divide this section in two subsections, the first one is

related to the select operator where we performed the same experimental protocol and ran the same lookup scenarios described in [4]. The second one is related to the update scenario where we performed the same experimental protocol and ran the same scenarios described in [2]. We implemented our data structure and performed all the experiments using the database cracking simulator[3] presented by [4]. We ran the experiments on a MacOS Sierra (10.12) machine with 2.2GHz quad-core Intel Core i7 processor (Turbo Boost up to 3.4GHz), 6MB shared L3 cache and 8 GB of RAM.



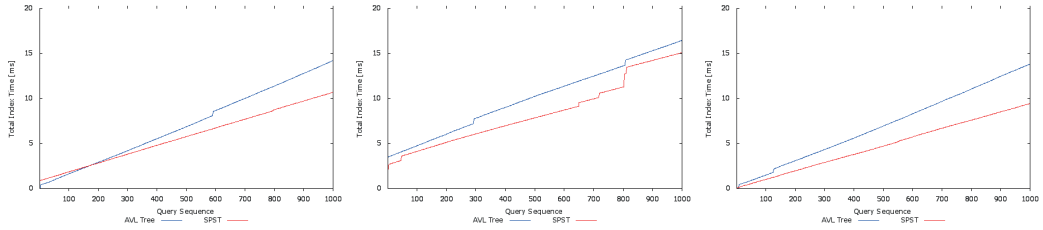
(a) Random (b) Sequential (c) Skewed

Figure 4. Workload Patterns

For the select operator, we focused our analysis on the accumulated index lookup time for querying and indexing, and the accumulated index update time. In particular, we analyzed the Instructions per Cycle (IPC) and the cache misses (L1/2/3). We consider as the best cache-efficient data structure the one with the highest IPC and lowest number of cache misses.

For the update operator, we considered two update scenarios: low frequency high volume updates (i.e, LFHV), and high frequency low volume updates (i.e, HFLV). In the first scenario after 1,000 queries a batch of 1,000 updates are executed. In the second scenario after 10 queries a batch of 10 updates are executed. The query pattern and the updates are both random. The SPST prunes itself always before a batch update if the previous queries present a standard deviation, for cracking time, lower than a defined threshold. We focused our analysis only on measurements that are affected by update and pruning (i.e, cracking time, index update time, cracker column shuffle time and pruning time).

We use an integer array with 10^8 uniformly distributed values. The workload size and the query selectivity is 1,000 and 1 for all experiments. All query predicates are of the form: $R.A \geq V_1$ AND $R.A < V_2$. We repeat the entire workload 5 times and take the average runtime of each query. We consider three different workloads depicted by Figure 4. For each workload, we graphically illustrate how a sequence of $1,000$ queries accesses the domain value of a single attribute. For each query, we plot the two edges of the interval (i.e., called "Query Predicate Sequence"). The random, sequential and skewed workloads are respectively depicted by Figures 4(a), 4(b), and 4(c). The skewed workload is generated by the zipf's law with α equals to 2.0.



(a) Random Workload (b) Sequential Workload (c) Skewed Workload

Figure 5. Sum of Lookup for Querying and Indexing, and Insertions Time in Various Workloads

Tree	L1	L2	L3	IPC
Random				
AVL	1108508606	4972838130	252404784	1.094
SPST	267097844	3957313535	135615510	1.385
Sequential				
AVL	855925856	10890330930	412469096	1.234
SPST	711228747	10479242239	399344564	1.263
Skewed				
AVL	573854301	3800678199	176536452	1.160
SPST	256760334	3780063118	128213328	1.600

Table 1. Cache Misses and IPC by workload

4.1. Select Operator

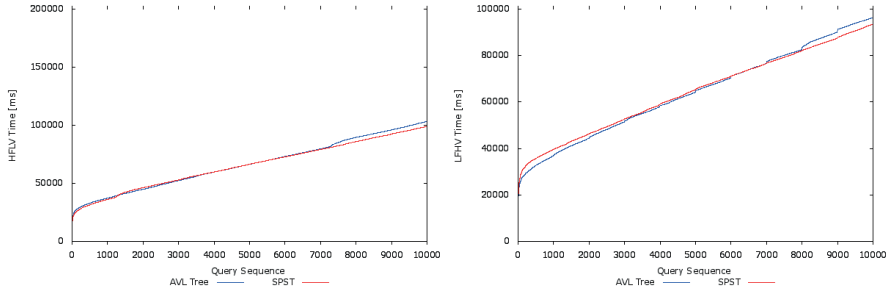
Figure 5 depicts the accumulated index lookup and maintenance time for the query stream in the random, sequential and skewed workload. For random, the AVL Tree was faster than the SPST for the first 180 queries, because the random workload demanded a higher number of rotations in the SPST to settle down the range pattern close to the root. With more incoming queries the SPST started to leverage the cached nodes from the root running the $1,000^{th}$ query 21.5% faster than the AVL Tree (see Figure 5(a)).

The sequential pattern was the worst case scenario for the SPST, but still the SPST was 7% faster than the AVL Tree at the $1,000^{th}$ query (see Figure 5(b)). The worst case scenario was the result of many changes in the range predicate of the sequential pattern that required splaying many nodes from the leaves. Over time the SPST mitigated these rotations with 16.9% less cache misses compared to the AVL (see Table 1). The skewed pattern was the best case scenario for the SPST, being 37% faster than the AVL Tree at the $1,000^{th}$ query (see Figure 5(c)). The best case scenario was the result of a skewed workload, achieving an IPC 37% higher. (see Table 1).

4.2. Update Operator

Figures 6(a) and 6(b) depicts the accumulated cracking and update time for the query stream of $10,000$ queries in the HFLV and LFHV scenarios respectively. In both, the SPST achieves the lowest run time. Every time the tree is pruned, updates are boosted but cracking becomes more expensive since we have less nodes to update, but bigger pieces of

[3]The cracker index simulator, written in C/C++ and compiled with G++ v.4.7, is available at: `www.infosys.uni-saarland.de/research/publications.php`



(a) HFLV (b) LFHV

Figure 6. Total time for cracking and updates

the cracker column to scan. The SPST was able to prune at convenient moments minimizing the extra cracking cost and greatly boosting update time. For HFLV, we defined empirically a standard deviation of 0.2 milliseconds and for LFHV 200 milliseconds. These values differ because for HFLV it is only analyzed the standard deviation for 10 queries previous to a batch update, while for LFHV 1,000 queries are analyzed. For HFLV, the SPST was pruned only once, and was 4% faster than the AVL Tree. For LFHV, the SPST was 5% faster, pruning the tree 8 times and having around 25% of the total size of a full AVL Tree. We observed more rotations in the SPST than in the AVL tree. However, the rotations in the SPST presented less impact in response time compared to the ones in the AVL Tree. While in the SPST the rotations happened most frequently near the root with less cache misses in L1/2/3 and higher IPC, the AVL Tree spanned many rotations usually close to the leaves of the index to rebalance the tree with many unnecessary tree nodes polluting the cache (see Table 1 cache misses).

5. Conclusion

This work presented the SPST as a cracker index for database cracking. We explored the Standard Cracking algorithm for select and mixed workloads with three different synthetic patterns where the SPST outperforms the AVL Tree in all scenarios. The SPST was able to cache the most frequently accessed data near to the root reducing cache misses and achieving a higher IPC than the AVL.

References

[1] J. Bell and G. Gupta. An evaluation of self-adjusting binary search tree techniques. *Software: Practice and Experience*, 23(4):369–382, 1993.

[2] S. Idreos, M. L. Kersten, and S. Manegold. Updating a cracked database. In *SIGMOD*, pages 413–424, 2007.

[3] S. Idreos, M. L. Kersten, S. Manegold, et al. Database cracking. In *CIDR*, volume 3, pages 1–8, 2007.

[4] F. M. Schuhknecht, A. Jindal, and J. Dittrich. The uncracked pieces in database cracking. *VLDB*, 7(2):97–108, 2013.

[5] D. D. Sleator and R. E. Tarjan. Self-adjusting binary search trees. *Journal of the ACM (JACM)*, 32(3):652–686, 1985.

3DR-Indexing: um Método para Identificação Automática dos Melhores Atributos de Indexação em Deduplicação de Dados

Levy de Souza Silva, Mirella M. Moro

[1]Departamento de Ciência da Computação - Universidade Federal de Minas Gerais

{levysouza,mirella}@dcc.ufmg.br

Resumo. *Nesta dissertação, propomos o método 3DR-Indexing para seleção de atributos relevantes com foco na etapa de indexação do processo de deduplicação de dados. Analisamos o impacto de tal seleção sobre o processo de deduplicação como um todo através de experimentos em dados reais e sintéticos de diferentes domínios. Os resultados mostram que nossa solução é eficiente (tempo de deduplicação) e eficaz (qualidade dos resultados) como um todo. Finalmente, o potencial de impacto deste trabalho está na contribuição para um problema essencial que aparece sempre que é necessário utilizar dados de fontes diferentes, incluindo várias aplicações Web, Big Data e Data Science.*

Abstract. *We propose the method 3DR-Indexing to select relevant attributes for the indexing step of the data deduplication process. We executed an extensive performance analysis and conclude that our solution is efficient (deduplication time) and effective (quality of results) as a whole. The potential impact of this work lies in contributing to an essential problem that appears whenever using data from different sources is necessary, including several Web, Big Data and Data Science applications.*

1. Introdução

Deduplicação de dados é o problema de identificar registros duplicados em fontes de dados. Registros duplicados são instâncias de dados que representam o mesmo objeto no mundo real. Este problema vem sendo estudado em ambas academia e indústria. Por exemplo, na biblioteca digital de Ciência da Computação – DBLP, existem pelo menos 1.005 artigos publicados sobre o assunto entre os anos de 2012 e 2018 (i.e., uma média de 143 estudos por ano). Considerando os anos anteriores de 2005 a 2011, há aproximadamente 258 estudos, ou seja, um aumento de quase 300% em sete anos.[1]

É um problema presente em várias aplicações reais, relevantes e complexas. Por exemplo, deduplicação ajuda a identificar produtos repetidos em lojas online [Barbosa et al. 2018], contatos duplicados em dispositivos móveis [Borges et al. 2017] e duplicatas em dados médicos, financeiros e governamentais [Konda et al. 2019]. Mais amplamente, em Recuperação de Informação, ela é importante para remover documentos duplicados dos resultados retornados por motores de busca, bibliotecas digitais e sistemas de indexação de texto (e.g., páginas web e citações bibliográficas) [Carvalho et al. 2011]. Em resumo, é uma tarefa essencial para as áreas de Limpeza e Qualidade de Dados, bem

[1]Encontrados na DBLP Computer Science Bibliography em 07/2019 - keywords: Deduplication, Record Linkage e Entity Resolution. URL: http://dblp.uni-trier.de

Tabela 1. Exemplo de identificação de registros duplicados em duas fontes de dados

CD-ID	ID	Fonte	Título	Artísta	Categoria	Gênero	Extra	Ano
A	1	Atlantic Records	The Moment	Kenny G	Jazz	Jazz	ID3G: 8	1996
	2	J Records	Moment	Kenny G	Null	Classical	Null	1996
B	3	Atlantic Records	Donde Hay Musica	Erros Ramazzotti	Jazz	Ballad	Null	Null
C	4	Atlantic Records	Alien Ant Farm	Anthology	Rock	Rock	Null	Null
	5	J Records	Alien Ant Farm	Anthology	NewAge	AlternRock	ID3G: 40	2001

como para qualquer problema/solução que considera grandes volumes de dados provenientes de fontes diferentes, como nos contextos de Web, Big Data e Ciência dos Dados.

O processo para identificar registros duplicados é composto de três principais etapas: (1) **indexação**, que atribui uma chave para cada registro; (2) **clusterização**, que agrupa as instâncias de acordo com o valor da chave; e (3) **classificação**, que classifica os registros pertencentes a cada grupo. Nosso foco é na etapa de *indexação*, que cria estruturas de Chave de Bloco (do termo em inglês *Block Key* - *BK*) para agrupar registros similares [Christen 2012]. A indexação é muito importante porque sem ela cada instância seria comparada com todas as outras, resultando em uma complexidade quadrática.

Abordagens de deduplicação usam BKs com diferentes propósitos, como ordenar e clusterizar os registros. Exemplos incluem *Standard Blocking* [Fellegi and Sunter 1969], *Sorted Neighborhood* [Hernández and Stolfo 1995], *Canopy* [McCallum et al. 2000] e *Adaptive Sorted Neighborhood* [Yan et al. 2007]. Porém, em todos os casos, o Valor de Chave de Bloco (do inglês *Block Key Value* - *BKV*) é definido de acordo com os atributos disponíveis. Idealmente, tais atributos de indexação devem ser eficazes (i.e., para melhor distinguir os registros) e eficientes (i.e., para permitir um rápido processo de deduplicação). A seguir, um exemplo é apresentado para justificar este trabalho, seguido por um sumário de correntes soluções e nossas contribuições.

Exemplo. A Tabela 1 exibe um exemplo com cinco instâncias de duas fontes de dados identificadas pela coluna *ID*. Tais instâncias são três CDs de música (coluna *CD-ID*). Assim, o objetivo é encontrar os registros duplicados nestas fontes de dados. Logo, na etapa de indexação, os registros são indexados por um atributo evitando comparar todos os 10 pares de registros. Então, dependendo do atributo de indexação escolhido, a seguinte situação pode ocorrer:[2] (*i*) indexando por *Fonte* são definidos um bloco para os registros de *Atlantic Records* e outro para *J Records* – CDs 1 e 2 não são comparados, o que prejudica a eficácia; (*ii*) indexando por *Ano* são criados dois blocos com CDs $\{1,2\}$ e $\{5\}$ – o processo compara CDs 1 e 2, mas falha em comparar 4 e 5; (*iii*) indexando por *Categoria*, *Gênero* e *Extra* é igualmente ineficaz devido às mesmas razões, e diferentes fontes podem ter distintas interpretações para categoria e gênero; (*iv*) indexando por *Artista* ou *Título* terá maior eficácia, pois o processo compara os CDs 1 e 2, bem como 4 e 5.

Soluções Atuais. O processo de seleção de atributos era puramente arbitrário [Hernández and Stolfo 1995]. Atualmente, ele é baseado no conhecimento de especialistas sobre os dados. Há pelo menos duas estratégias para definir as BKs: *schema-agnostic*, que utiliza o valor do atributo como chave; e *schema-based configurations*, que combina regras extraídas dos valores dos atributos para criar as BKs [Papadakis et al. 2015]. Em

[2]Utilizando o próprio valor do atributo como chave de bloco.

ambas, usuários especialistas são necessários para escolher os atributos. Com o *3DR-Indexing*, os atributos são selecionados automaticamente. Ademais, a maioria dos estudos seleciona atributos relevantes para a etapa de *classificação*, onde o objetivo é escolher atributos para comparar as instâncias. Soluções incluem algoritmos de aprendizagem de máquina [Chen et al. 2012] e estratégias com informações extraídas dos dados [Canalle et al. 2017]. Porém, não existe solução adaptada para a etapa de *indexação*.[3]

Principais Contribuições. Nós propomos o método *3DR-Indexing* que seleciona atributos relevantes para a etapa de indexação do processo de deduplicação. Analisamos o impacto de tal seleção sobre o processo de deduplicação como um todo e a combinação do 3DR com algoritmos do estado da arte em uma ampla avaliação experimental, que contempla vários domínios de dados (bibliografia, música, pessoal e restaurantes). Todo o código criado e os conjuntos de dados definidos estão publicamente disponíveis[4]. Finalmente, os resultados desta dissertação estão publicados em: um artigo completo [Silva et al. 2017] e um artigo curto [Silva and Moro 2017] no Simpósio Brasileiro de Bancos de Dados (principal evento nacional da área), e um artigo completo [Souza et al. 2018] no 12th *Alberto Mendelzon Int'l Work. on Foundations of Data Management*, provavelmente o evento internacional mais relevante de fundamentos de bancos de dados fora da hegemonia ACM-IEEE-SBC.

2. Processo de Deduplicação de Dados

Seja D um conjunto de dados contendo *i* registros, tal que $D = \{r_1, r_2, .., r_i\}$. Cada registro r é definido por um conjunto de atributos $A_r = \{a_1, a_2, .., a_j\}$. Assim, o processo de deduplicação de dados é composto de três principais etapas, como segue.

Indexação. Nesta etapa, o objetivo é associar um valor de chave de bloco para cada registro. Primeiro, um atributo é escolhido do conjunto disponível. Depois, o BKV é definido como o valor do atributo ou uma codificação é aplicada sobre o valor do atributo. Depois da indexação, cada $r \in D$ está associado a um BKV. Uma técnica popular para codificação é o algoritmo *Soundex* [Christen 2012].

Clusterização. Após a indexação, os registros são agrupados baseado no valor de chave de bloco. Um algoritmo popular é o *Standard Blocking* que cria um conjunto de blocos onde cada bloco agrupa registros similares. Assim, os registros são comparados apenas dentro de cada bloco (criado de acordo com o *BKV* de cada registro). Outra opção é o *Sorted Neighborhood* que combina os registros por meio de uma chave ordenada (similar a uma *BK*). Porém, antes de realizar as comparações, os registros são ordenados de acordo com a chave. Depois, uma janela deslizante de tamanho $w > 1$ percorre todos os registros de D, e o primeiro registro da janela é comparado com todos os outros da mesma janela.

Classificação. Por fim, para avaliar quão similares dois valores são, uma função de similaridade calcula a correspondência entre os atributos e retorna um valor no intervalo $[0, 1]$, onde 1 é a similaridade máxima. Algoritmos populares são *Jaro* e *Jaro Winkler* [Christen 2012]. No fim desta etapa, os registros são classificados como duplicados, não duplicados e possíveis duplicados, baseado em um limiar de similaridade definido.

[3]Outras discussões sobre trabalhos relacionados são apontadas na Seção 3.6 do texto da Dissertação.

[4]`http://www.dcc.ufmg.br/~mirella/projs/deduplica`

3. Identificação dos Melhores Atributos de Indexação

Esta seção apresenta brevemente o método *3DR-Indexing*, que cria um *ranking* para cada atributo $a \in A_r$ baseado em uma combinação de métricas.

Definição das Métricas. Considere *notNull* o número de valores não nulos e válidos para um atributo *a*, *dupValues* o número de valores duplicados para *a*, *distValues* o número de valores distintos para *a*, e T o número total de instâncias do conjunto de dados. Então, as Equações 1, 2, 3 e 4 definem respectivamente quatro métricas nas quais o *3DR-Indexing* é baseado: (1) **Duplicidade**, que considera a divisão entre os valores duplicados e o total de valores não nulos; (2) **Distintividade**, que consiste da fração entre os valores distintos e o total de valores não nulos; (3) **Densidade**, que considera a fração entre o número de valores não nulos e o total de instâncias; e (4) **Repetição**, que consiste da divisão entre o total de valores repetidos e o total de valores distintos. As métricas são normalizadas por seu valor máximo em A_r e denotadas por $Dup(a)$, $Dist(a)$, $Dens(a)$ e $Rep(a)$.

$$\overline{\mathrm{Dup}}(a) = \frac{\mathrm{dupValues}(a)}{\mathrm{notNull}(a)} (1) \quad \overline{\mathrm{Dist}}(a) = \frac{\mathrm{distValues}(a)}{\mathrm{notNull}(a)} (2) \quad \overline{\mathrm{Dens}}(a) = \frac{\mathrm{notNull}(a)}{T} (3) \quad \overline{\mathrm{Rep}}(a) = \frac{T - \mathrm{distValues}(a)}{\mathrm{distValues}(a)} (4)$$

Tais métricas podem ser facilmente computadas em qualquer conjunto de dados relacional, e extensíveis para NoSQL. De fato, elas são comumente apresentadas em histogramas de sistemas de bancos de dados. Assim, o *3DR-Indexing* é adaptável para vários domínios. As métricas *Densidade* e *Repetição* também são utilizadas em outros estudos para selecionar atributos relevantes com foco na etapa de classificação [Canalle et al. 2017] (mas esta é a primeira vez na etapa de indexação).

O Método *3DR-Indexing*. Nossa solução define um *score* para cada atributo baseado em seus valores de *Densidade*, *Duplicidade*, *Distintividade* e *Repetição*. O objetivo é melhorar o *trade-off* entre eficiência e eficácia do processo de deduplicação de dados, ou seja, selecionar os atributos de melhor eficácia que executem o processo de deduplicação mais rápido. Assim, a relevância não normalizada do atributo a é definida como:

$$\bar{R}(a) = \mathrm{Dens}(a) + \mathrm{Dup}(a) + ((1 - \mathrm{Dist}(a)) \times \mathrm{Dens}(a)) + (1 - \mathrm{Rep}(a)). \quad (5)$$

As métricas *Densidade* e *Duplicidade* melhoram a eficácia pois atributos densos e duplicados geram blocos mais eficazes. Por outro lado, alta *Repetição* e *Distintividade* produzem tempo elevado porque criam poucos blocos com muitos registros (i.e., muitas comparações) ou muitos blocos com poucos registros (i.e., muitos acessos ao banco para selecionar os registros de cada bloco). Logo, utilizamos o complemento destas métricas para melhor eficiência, pois atributos sem muita *Repetição* e *Distintividade* executam a deduplicação mais rápido. Ademais, multiplicamos $(1 - Dist(a))$ pela Dens(a) para evitar que baixa *Distintividade* e *Densidade* gerem altos *scores*; ou seja, consideramos a *Densidade* de cada atributo como peso para a *Distintividade* do mesmo[5]. Por fim, o *3DR* é combinado com os principais algoritmos de deduplicação (Seção 4.4 desta dissertação).

4. Metodologia Experimental

Os experimentos são divididos em cinco questões: **(Q1)** Qual parte possui maior impacto no processo de deduplicação entre a função de indexação, o atributo de indexação e o algoritmo de clusterização? **(Q2)** O mesmo atributo de indexação é eficiente e eficaz nos

[5]Mais discussões sobre as métricas são apontadas na Seção 6.5 do texto da Dissertação.

métodos *Schema-Agnostic* e *Configurations-Based*? **(Q3)** Os resultados da deduplicação são eficazes utilizando o mesmo atributo de indexação nos algoritmos *Standard Blocking* e *Sorted Neigborhood*? **(Q4)** O método *3DR-Indexing* encontra os melhores atributos para a etapa de indexação? **(Q5)** Os resultados da deduplicação são melhorados utilizando uma combinação de atributos na etapa de indexação? Assim, os experimentos consistem em executar a deduplicação com diferentes *setups* de indexação, conforme atributos disponíveis no *dataset*. Ademais, os experimentos consideram eficiência e eficácia. A *F-Measure* é adotada na eficácia. Para a eficiência, nós calculamos o tempo para recuperar os valores de BK distintos, as tuplas de cada BK e executar o processo de deduplicação.

Conjuntos de Dados. Nós utilizamos 15 *datasets* entre reais e sintéticos com vários contextos e tipos de atributos. Os sintéticos são criados com o *Data Set Generator Program* [Christen 2012], e os reais com o *DuDe toolkit*[6]: *CORA*, *Restaurant* e *CD Information*; os quais são amplamente utilizados em experimentos de deduplicação.

5. Resultados e Discussões

Os resultados desta dissertação cobrem vários *setups* experimentais incluindo dados reais e sintéticos, variação no total de duplicatas (10% a 90%) e no total de instâncias (10^2 a 10^6), algoritmos de clusterização *Standard Blocking* e *Sorted Neighborhood* e métodos de indexação *schema-agnostic* e *configurations-based* (Seção 5.1 desta dissertação).

No geral, o atributo de indexação tem maior efeito sobre o processo de deduplicação, e a escolha do atributo adequado é fundamental para uma deduplicação eficaz e eficiente (**Q1**). Em média, os melhores atributos diferem dos piores em 44% na eficácia, ou seja, a escolha aleatória ou sem critério faz com que cerca de 44% das duplicatas não sejam identificadas. Além disso, comparando o efeito do atributo de indexação, da função de indexação e do algoritmo de clusterização sobre o processo, o atributo de indexação tem maior efeito (Seções 6.1 e 6.5 desta dissertação).

Considerando *Schema-Agnostic vs. Configurations-Based*, os atributos mais eficazes mantêm a eficácia em ambos os métodos (**Q2**). Ademais, a função de indexação não influencia os resultados experimentais porque os valores de eficácia são semelhantes em ambos métodos. Observando *Standard Blocking vs. Sorted Neighborhood*, os resultados são semelhantes, pois os atributos mais eficazes mantêm a eficácia em ambos os algoritmos (**Q3**). Mas, atributos com alta *Repetição* são eficazes apenas no algoritmo de blocos, pois todos os registros repetidos estão no mesmo bloco (Seção 6.2 desta dissertação).

Analisando o *3DR-Indexing*, os atributos mais relevantes têm o melhor *trade-off* entre eficiência e eficácia na deduplicação (**Q4**). No geral, o *3DR* identifica os melhores atributos em 10 de 13 *datasets*, e a combinação de métricas proposta supera outras combinações/*baselines* (Seção 6.3.5 desta dissertação). Mesmo quando o *3DR* falha, ou seja, não seleciona o melhor atributo, os outros atributos de maior relevância ainda têm resultados significantes em termos de eficácia e tempo de execução. Ademais, o *3DR* supera os resultados de *seis* trabalhos do estado da arte (Seção $6.3.6$ desta dissertação).

Por fim, os experimentos também avaliam a combinação de atributos na indexação (**Q5**). No geral, os resultados não têm aumento significante quando uma combinação de atributos é utilizada na indexação, pois o valor de eficácia é semelhante ao conseguido

[6] https://hpi.de/naumann/projects/data-quality-and-cleansing/dude-duplicate-detection

com apenas um atributo. Além disso, o tempo médio da deduplicação é 15 vezes maior quando uma combinação de atributos é utilizada (Seção 6.4 desta dissertação).

6. Considerações Finais

Nós apresentamos o método *3DR-Indexing* que seleciona atributos relevantes para a etapa de indexação da deduplicação e avaliamos o impacto da escolha do atributo de indexação sobre as outras etapas do processo. A principal conclusão é que os resultados com melhor *trade-off* de eficiência e eficácia são obtidos com os atributos melhores ranqueados pelo *3DR-Indexing*. No futuro, nós planejamos avaliar o método proposto em novos domínios.

Agradecimentos. Trabalho parcialmente financiado por FAPEMIG e CNPq.

Referências

Barbosa, L. et al. (2018). Big data integration for product specifications. *Technical Committee on Data Engineering*, 41(2):71–81.

Borges, E. N. et al. (2017). Contact deduplication in mobile devices using textual similarity and machine learning. In *ICEIS*, pages 64–72, Porto, Portugal.

Canalle, G. K. et al. (2017). A strategy for selecting relevant attributes for entity resolution in data integration systems. In *ICEIS*, pages 80–88, Porto, Portugal.

Carvalho, A. P. et al. (2011). Incremental unsupervised name disambiguation in cleaned digital libraries. *JIDM*, 2(3):289–304.

Chen, J. et al. (2012). A learning method for entity matching. In *QDB*, Istanbul, Turkey.

Christen, P. (2012). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, Berlin.

Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Hernández, M. A. and Stolfo, S. J. (1995). The merge/purge problem for large databases. In *ACM SIGMOD*, pages 127–138, New York, USA.

Konda, P. et al. (2019). Executing entity matching end to end: A case study. In *EDBT*, pages 489–500, Lisbon, Portugal.

McCallum, A. et al. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In *ACM SIGKDD*, pages 169–178, Boston, USA.

Papadakis, G. et al. (2015). Schema-agnostic vs schema-based configurations for blocking methods on homogeneous data. *VLDB Endowment*, 9(4):312–323.

Silva, L. S. et al. (2017). Uma avaliação de eficiência e eficácia da combinação de técnicas para deduplicação de dados. In *SBBD*, pages 160–171.

Silva, L. S. and Moro, M. M. (2017). Análise do impacto do gerador de conjuntos de dados em experimentos de deduplicação de dados. In *SBBD*, pages 228–233.

Souza, L. et al. (2018). Automatic identification of best attributes for indexing in data deduplication. In *AMW*, CEUR Workshop Proceedings 2100, paper 14.

Yan, S. et al. (2007). Adaptive sorted neighborhood methods for efficient record linkage. In *JCDL*, pages 185–194, Vancouver, Canada.

Privacy-Preserving Attribute Pairing

Thiago Pereira da Nóbrega[1], **Carlos Eduardo Santos Pires**[1]

[1]Programa de Pós-graduação em Ciência da Computação
Universidade Federal de Campina Grande (UFCG) – Campina Grande, PB

thiago.pereira@copin.ufcg.edu.br, cesp@dsc.ufcg.edu.br

Abstract. *Privacy-Preserving Record Linkage (PPRL) aims to identify entities, stored in different databases, that correspond to the same real-world object while preserving the privacy of entities during the linkage process. PPRL can be employed in several contexts where privacy is an issue. For instance, to identify patients who have the same pathology (e.g. breast cancer) in different hospitals in order to elaborate a machine learning model for early detection of the pathology. The first step of PPRL is the agreement of the parties about the data (attributes) that will be used during the record linkage process. In order to reach an agreement, the parties must share information about their data schema, which in turn can be utilized to break the data privacy. To overcome the (vulnerability) problem caused by the schema information sharing, we propose a novel privacy-preserving approach for attribute pairing to aid PPRL applications. Empirical experiments show that our approach improves considerably the efficiency and effectiveness in comparison to a state-of-the-art baseline.*

1. Introduction

Integrating data from multiple sources has been a traditional challenge for the database community during the last decades. This challenge gets hampered when we need to integrate private data utilizing techniques such as Record Linkage (RL), i.e., the task of identifying and linking records that correspond to the same real-world object. Such challenge claims for sophisticated RL approaches capable of preserving the privacy of data during the integration process. Such approaches are proposed to address the Privacy-Preserving Record Linkage (PPRL), i.e., the task of performing RL over data sources that belong to different parties (suppliers) without revealing any information that could compromise the privacy of the parties' data [Vatsalan et al. 2013]. The applicability of PPRL approaches include hospitals that want to investigate disease outbreaks, banks that seek to reduce credit fraud, and national security applications [Vatsalan et al. 2019].

In a PPRL process, the parties involved must firstly agree on the entities that will be matched/linked during the task processing. After that, each party converts, independently, the information (data and metadata) regarding its records into a single-table schema. The single-table schemas (of each party) must be aligned, through attribute pairing, in order to allow the identification of correspondences between the attributes. For example, a hospital could provide a single-table schema that represents patients with cancer (records) including the attributes *name*, *address*, and *birth date*. Analogously, another hospital could provide a different single-table schema representing its patients with the following attributes *fullname*, *domicile*, and *age*. In traditional RL, where privacy is not a requirement, it is simple to identify the correspondences according to the attribute name

(or content). However, in PPRL, such information is often unavailable due to privacy constraints imposed by the parties.

According to recent PPRL surveys [Vatsalan et al. 2013, Vatsalan et al. 2018, Franke et al. 2019, Vatsalan et al. 2019], most of the state-of-the-art approaches require that the parties disclose some information (e.g., data type and attribute names) about their respective single-table schema. However, such information can be inadvertently used to break the secrecy and privacy of the data. For instance, consider a data source that stores information about a medical clinical trial. This data source contains anonymized information about the subjects (e.g., *names*, *age*, and *zipcode*) and the tests performed in the subjects (e.g., amount of a drug component in subject's organs). Thus, if the data source schema is revealed to an adversary, it could identify drug components according to the tests performed in the subjects. Moreover, using the semantics of the attributes, the adversary could execute attacks over the anonymized data [Christen et al. 2017].

To address the problem of disclosing information about the single-table schema, in this paper, we propose the Privacy-Preserving Attribute Pairing (PPAP), a novel semi-automatic privacy-preserving approach for attribute pairing to be employed in a semi-honest adversary model[1]. BAP intends to increase the privacy of the PPRL task by executing attribute pairing without the need of parties to disclose information about their schemas or data. Instead of using the anonymized data source or the attribute names, BAP represents each attribute of a data source as a data signature. The signatures are used by a trusted third-party to perform attribute pairing. The use of signatures enables BAP to: i) increase the privacy of PPRL by reducing the amount of information shared by the parties; and ii) minimize the computational cost of BAP by limiting the generation of signatures to a small portion of the data source (sample). In summary, the main contributions of this work are: i) a semi-automatic privacy-preserving attribute pairing approach, using a three-party protocol[2], under the semi-honest adversary model that prevents participants from learning about the schema or data of the other participants and ii) an empirical evaluation of PPAP using real-world data.

2. BACKGROUND

In this section, we introduce the building blocks required to understand the PPAP approach.

2.1. Data Anonymization

In order to guarantee the privacy of the original data in PPRL, it is necessary to use data anonymization techniques in such a manner that: i) masked data cannot be mapped to the original data; and ii) when calculating the similarity between two masked data, the result must be the same when calculating the similarity between their respective original values [Vatsalan et al. 2013].

The most common anonymization technique employed in PPRL is Bloom filter [Schnell et al. 2009]. A filter consists of an array of n-bits (filter length). Initially, all bits are set to '*0*'. Then, the original data is converted into a set of *substrings* (*q-grams*)

[1] A semi-honest adversary model is a model in which all parties follow the protocol honestly but will try to find out as much as possible about each other's data.

[2] Three-party protocols, use a (trusted) third-party to conduct the attribute pairing from two data sources.

and inserted into the array through the application of hash functions. The output of such functions indicate the positions of the array that must be changed to represent the *substrings*. The similarity of two filters can be calculated using Token distance functions such as $DICE = \frac{2\times|a\cap b|}{|a|+|b|}$, where $|a \cap b|$ is the number of positions with the value *1* that coincide in the two filters; and $|a|$ and $|b|$ represent the total number of 1s in each filter.

2.2. PPRL

The basic idea of PPRL is to execute the linkage process in anonymized data (by perturbating the original data with the use of encryption, hash functions, and noise additions), ensuring that the privacy and confidentiality of the data are preserved during the linkage process. PPRL reveals only a limited amount of information. For instance, a party only knows which of its own records exist in the data source of the other party [Vatsalan and Christen 2016]. Basically, the PPRL process includes the following steps [Vatsalan et al. 2013]:

- *Data pre-processing*: converts the original data into a standard format, previously agreed between the parties;
- *Data anonymization*: anonymizes the original data;
- *Blocking* (or filtering): reduces the number of comparisons by pruning the records that unlikely correspond to matches;
- *Comparison*: applies similarity (or comparison) functions to the anonymized data in order to calculate the similarity between candidate pairs;
- *Classification*: receives the calculated similarity values of the candidate pairs and classifies each of them into similar, dissimilar, or potentially similar;
- *Evaluation*: evaluates the efficiency (i.e., execution time, memory consumption, and CPU usage), effectiveness (measured by quality metrics such as Precision, Recall, and F-measure) and privacy of the PPRL process (measured by disclosure risk and information gain).

Most PPRL techniques suggest an implicit step (denoted here as *Handshake*) that occurs prior to the *Data pre-processing* step [Vatsalan et al. 2013]. In the *Handshake* step, besides agreeing on data anonymization parameters (e.g., cryptography keys and Bloom filters length), the parties explicitly agree on the attributes that will be used to match/link the records during the following steps (*Blocking*, *Comparison*, and *Classification*) of PPRL. However, to perform attribute pairing, the parties have to disclose information about their schemas. Such information can be used to break the privacy of the data. Thus, our work aims to improve the *Handshake* step by proposing a semi-automatic approach to eliminate the need for the parties to disclose information about their schemas.

2.3. Data Signature

Data signature is the representation of an existing characteristic of a large data source. Such representation is usually much shorter than the original information [Ahmadi et al. 2009]. In a privacy-preserving context, a signature can be employed to represent a characteristic of an attribute (e.g., the cardinality of a schema attribute) and hide information that could be used to break the privacy of the original information. For instance, by representing the cardinality of an attribute as a signature, we can hide the attribute values from the other parties. In this work, we employ two signatures for each attribute: i) a representation of all values assumed by the attribute in the data source, and ii) the amount of information represented by the attribute.

3. Privacy-Preserving Attribute Pairing Approach

In this section, we describe the Privacy-Preserving Attribute Pairing (PPAP)[3] approach. The main goal of PPAP is to identify attribute correspondences between two single-table schemas. The PPAP approach is divided into three phases: *Sampling and Data Preprocessing*, *Signature Generation*, and *Private Attribute Pairing*.

Figure 1 illustrates the PPAP execution. Initially, in the *Sampling and Data Preprocessing* phase (Step 1), each party extracts a random sample from its single-table schema data source, converts that sample into a standard format and anonymizes the sample. Next, in the Signature Generation phase (Step 2), the signatures are generated considering two characteristics presented in the anonymized sample to represent each attribute: i) the amount of information stored by each attribute (using the Shannon Entropy), and ii) the representation of all anonymized values of an attribute (using the Minhash representation). It is worthwhile to mention that both characteristics present strong privacy-preserving guarantees [Yan et al. 2017].

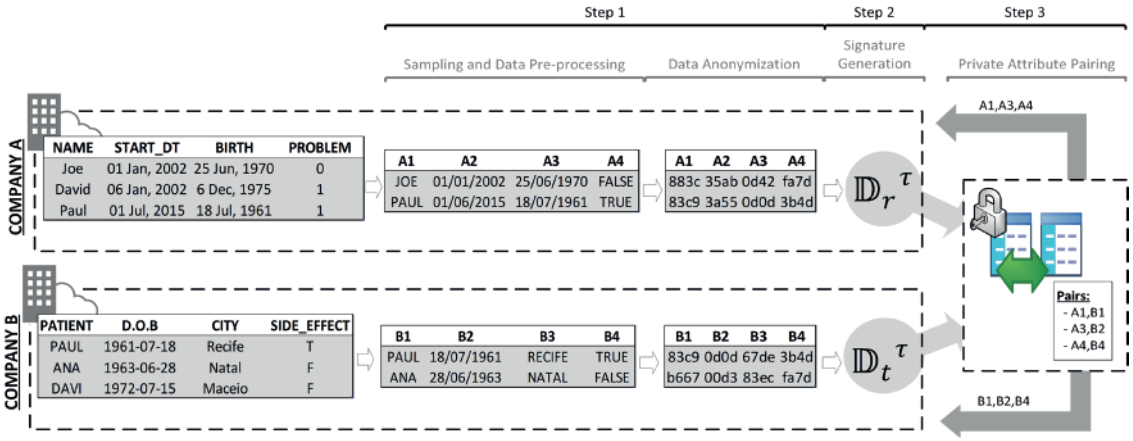


Figure 1. Outline of PPAP. First, each participant takes a sample, converts, anonymizes and generates a pair of signatures for each attribute in its data source. The signatures are sent to a trusted third-party, which matches the attributes.

After the *Signature Generation* phase, the signatures are sent to a trusted third-party which identifies the attribute correspondences, *Private Attribute Pairing* phase (Step 3). Finally, the trusted third-party informs to each party which attributes of their single-table schema will be used in the following steps of PPRL (i.e., *Blocking*, *Comparison*, and *Classification*). Thus, the party can perform the PPRL without disclosing information about the schema or data to the other party.

4. Main results

We evaluate the efficiency, effectiveness, and the privacy of the PPAP approach using eight different real-world data sources. Figure 2 shows the effectiveness (quality of attribute pairing) of our approach; the vertical axis represents F-measure (higher values are better) while the horizontal axis denotes the similarity threshold employed. The continuous colored line shows the quality achieved by the PPAP approach using different sample

[3]Available from https://github.com/thiagonobrega/BAP

strategies and the red dashed line illustrates the quality reached by an approach that does not consider the privacy of the data during the pairing [Bilke and Naumann 2005].

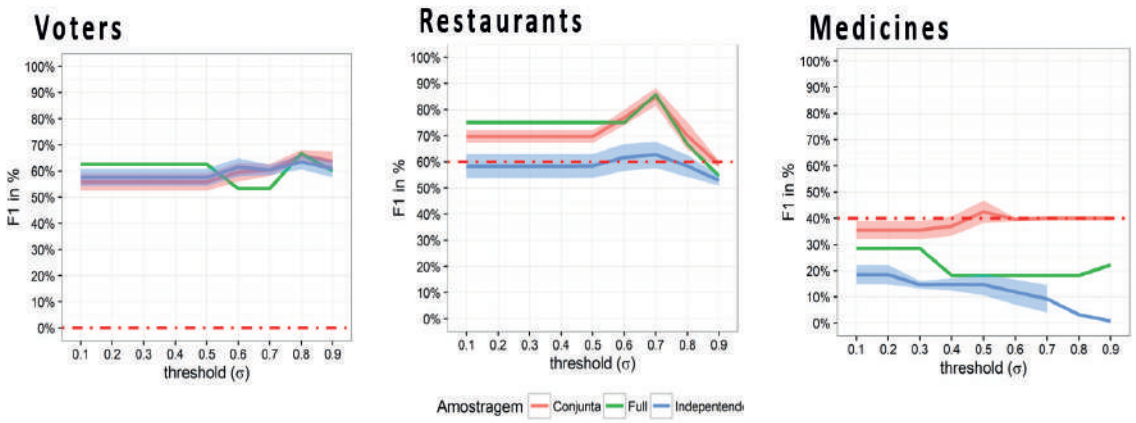


Figure 2. Quality of the attribute pairing

It is worthwhile to mention that our approach achieves higher quality and efficiency (execution time) results than the competitors. These results are detailed in the dissertation [Nóbrega 2018]. In particular, Chapters 4 and 5 offer a clear explanation of the proposed approach and the experimental results, respectively.

5. Conclusion

In this work, we have presented a novel privacy-preserving approach to perform attribute pairing in two single-table schemas of different data sources without revealing private information to the parties. The approach employs data signatures to identify attribute correspondences using a trusted third-party. The evaluation has shown that our approach achieves better effectiveness and efficiency results than the competitors, including competitors that do not consider the privacy during attribute pairing.

It is worthwhile to mention that the results achieved by this work are aligned with the restriction imposed by the majority of the data privacy laws, i.e., Brazilian *"Lei Geral de Proteção de Dados"* (LGPD) and the European *"General Data Protection Regulation"* (GDPR). In other words, given the restrictions to operate/manipulate private data imposed by laws and regulation, the PPAP approach can be employed to enable the usage of a wide range of privacy-preserving applications, such as PPRL, Privacy-Preserving Data Mining, Privacy-Preserving Machine Learning, and so on.

The products derivate from the dissertation are:

1. **Two conference papers**

QUALIS A1 Nóbrega, T. P., Pires, C. E. S., Araújo, T. B., Mestre, D. G. (2018). Blind attribute pairing for privacy-preserving record linkage. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing - SAC '18 (pp. 557–564). Pau, France: ACM Press. `https://doi.org/10.1145/3167132.3167193`

QUALIS B2 Nóbrega, T. P. da, Pires, C. E. S., Araujo, T. B. (2016). Avaliação Empírica de Técnicas de Comparação Privada Aplicadas na Resolução de Entidades.

In Proceedings of the 31 st of the Brazilian Symposium on Databases (SBBD16) (pp. 121–126). Retrieved from `http://sbbd2016.fpc.ufba.br/e-book/proceedings.pdf`

2. **Awards**
   - i Brazilian Symposium on Databases (SBBD16): "*Best short paper Honourable Mentions*"
3. **Source code with the instruction to re-execute the experiments**
   - i `https://github.com/thiagonobrega/BAP`
4. **Datasets and gold standards**
   - i Datasets: `https://github.com/thiagonobrega/BAP/tree/master/execution`
   - ii Gold standards: `https://github.com/thiagonobrega/BAP/blob/master/confs/gabaritos.xlsx`

References

Ahmadi, B., Hadjieleftheriou, M., Seidl, T., Srivastava, D., and Venkatasubramanian, S. (2009). Type-based categorization of relational attributes. In *Proceedings of EDBT '09*, page 84, New York, New York, USA. ACM Press.

Bilke, A. and Naumann, F. (2005). Schema Matching Using Duplicates. In *21st International Conference on Data Engineering (ICDE'05)*, pages 69–80. IEEE.

Christen, P., Schnell, R., Vatsalan, D., and Ranbaduge, T. (2017). *Efficient Cryptanalysis of Bloom Filters for Privacy-Preserving Record Linkage Peter*, volume 10235 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham.

Franke, M., Gladbach, M., Sehili, Z., Rohde, F., and Rahm, E. (2019). ScaDS Research on Scalable Privacy-preserving Record Linkage.

Nóbrega, T. P. (2018). Pareamento privado de atributos no contexto da resolução de entidades com preservação de privacidade.

Schnell, R., Bachteler, T., and Reiher, J. (2009). Privacy-preserving record linkage using Bloom filters. *BMC Medical Informatics and Decision Making*, 9(1):41.

Vatsalan, D., B, D. K., and Gkoulalas-divanis, A. (2019). *An Overview of Big Data Issues in Privacy-Preserving Record Linkage*, volume 2. Springer International Publishing.

Vatsalan, D. and Christen, P. (2016). Privacy-preserving matching of similar patients. *Journal of Biomedical Informatics*, 59(December):285–298.

Vatsalan, D., Christen, P., and Verykios, V. S. (2013). A taxonomy of privacy-preserving record linkage techniques. *Information Systems*, 38(6):946–969.

Vatsalan, D., Karapiperis, D., and Verykios, V. S. (2018). Privacy-Preserving Record Linkage. (January).

Yan, Z., Liu, J., Li, G., Han, Z., and Qiu, S. (2017). PrivMin: Differentially Private MinHash for Jaccard Similarity Computation.

Abordagens de Aprendizado Ativo para Recuperação e Classificação de Imagens*

Rafael S. Bressan[1], **Pedro H. Bugatti (Co-orientador)**[1], **Priscila T. M. Saito (Orientadora)**[1]

[1]Departamento de Computação – Universidade Tecnológica Federal do Paraná (UTFPR) – Cornélio Procópio – PR – Brasil

{rafaelbressan}@alunos.utfpr.edu.br, {pbugatti, psaito}@utfpr.edu.br

Resumo. *Atualmente, bancos de dados de imagens vêm crescendo continuamente. Tal fato leva à necessidade de otimização e de aceleração nos processos de recuperação e de classificação de imagens, em conjunto com a melhoria na qualidade dos resultados retornados. Neste contexto, este trabalho propõe a utilização de estratégias de aprendizado ativo para recuperação e classificação de imagens, de forma a selecionar amostras mais informativas e minimizar a interação do especialista durante o processo de aprendizado. Além disso, novas estratégias de aprendizado de descritores e de aprendizado ativo são propostas para as tarefas de classificação e de recuperação de imagens baseada em conteúdo. Para validação das propostas, foram realizados experimentos utilizando conjuntos de dados de diferentes domínios de aplicação. A partir dos resultados obtidos, é possível observar ganhos significativos apresentados pelas propostas em relação às estratégias amplamente utilizadas na literatura.*

1. Introdução

Atualmente, bases de dados complexos (imagens, áudios, vídeos, entre outros) tornam-se cada vez maiores com os avanços nos dispositivos de aquisição e de armazenamento de dados. O desenvolvimento de abordagens automáticas de recuperação e de classificação tornou-se necessário para o gerenciamento e a organização desses dados. Considerando que: (i) existe uma grande quantidade de imagens não rotuladas disponíveis; (ii) o processo de anotação é crucial para o aprendizado efetivo da informação relacionada às imagens; (iii) a anotação manual é inviável em grandes conjuntos de dados, torna-se essencial o desenvolvimento de métodos robustos para lidar com essas questões.

Neste contexto, técnicas de aprendizado ativo têm sido propostas na área de *classificação* [Settles 2012]. Tais técnicas objetivam selecionar uma pequena quantidade de amostras mais informativas para a anotação do especialista e para o aprendizado do classificador. Em se tratando da área de *recuperação de imagens por conteúdo* (*Content-Based Image Retrieval* - CBIR), métodos embasados em realimentação de relevância têm sido explorados para amenizar problemas intrínsecos (e.g. gap semântico). Tais métodos possibilitam o refinamento da consulta inicial realizada por determinado especialista, visando torná-la mais precisa, retornando somente respostas relevantes e descartando as irrelevantes. Neste sentido, trabalhos na literatura [Rao et al. 2018, Wang et al. 2015] têm

*CNPq (grants #431668/2016-7, #422811/2016-5); CAPES; Fundação Araucária; SETI; e UTFPR.

explorado a combinação de técnicas de aprendizado ativo e técnicas de recuperação por conteúdo. No entanto, embora amplamente utilizadas e de certa forma bem sucedidas em diferentes domínios, muitas estratégias de aprendizado ativo, consideradas nesses trabalhos, tornam-se inviáveis. Por exemplo, considerando o contexto médico e as restrições apresentadas pelo mesmo (i.e. relacionadas a lidar com grandes conjuntos de dados, tempos de resposta interativos e intervenção mínima do especialista no processo de aprendizado).

Sendo assim, apesar de alguns esforços, ainda existem necessidades de otimizações, de forma a prover melhorias no processo de aprendizado e de recuperação das imagens. Portanto, o presente trabalho relacionado à dissertação de mestrado [Bressan 2018], visou diminuir essa lacuna, propondo técnicas de aprendizado mais efetivas e eficientes. As estratégias de aprendizado ativo propostas, diferentemente dos trabalhos na literatura, selecionam as imagens mais informativas considerando não apenas a similaridade com uma dada imagem de consulta, mas também critérios baseados em diversidade e incerteza (i.e. imagens a partir de diferentes classes e difíceis de serem classificadas, respectivamente). Tais critérios possibilitam acelerar e melhorar os processos de aprendizado e de recuperação de imagens.

2. Contribuições da Dissertação

As principais contribuições da dissertação consistem no desenvolvimento de técnicas de aprendizado mais robustas (i.e. eficazes e eficientes) para as áreas de recuperação [Bressan et al. 2019, Bressan et al. 2018c, Bressan et al. 2018a] e de classificação de imagens [Bressan et al. 2018b, de Souza Junior et al. 2018, Camargo et al. 2017]. Diferentemente dos esforços apresentados na literatura, as técnicas de aprendizado propostas levam em consideração aplicações que apresentam determinadas restrições de tempo e de recursos computacionais.

Para a tarefa de classificação, duas principais contribuições são propostas (destacadas na Figura 1). A primeira, uma metodologia para **avaliação de desempenho do aprendizado de classificadores** em diferentes conjuntos biomédicos [Camargo et al. 2017]. Nesta proposta são consideradas regras de relação de compromisso na escolha do melhor modelo de classificação. O cálculo de compromisso leva em consideração tempo computacional e acurácia. Tal cálculo é dado pela média harmônica (Equação 1), em que μAcc_{ij} refere-se à acurácia média; μT_i corresponde ao tempo computacional médio; i representa o i-ésimo conjunto de dados e j corresponde ao j-ésimo modelo de aprendizado supervisionado.

Dentre os classificadores supervisionados considerados (k-NN, MLP, OPF, CSVM, NuSVM, RF), o classificador OPF foi o modelo mais eficaz e eficiente, apresentando os melhores resultados (menores valores de médias harmônicas) em todos os conjuntos avaliados (ou seja, apresentou elevadas acurácias de classificação e menores tempos de treinamento e de teste). A metodologia proposta mostra-se como uma contribuição relevante, dado que possibilita analisar o desempenho de cada modelo de classificação em termos de escalabilidade, bem como determinadas restrições de tempo e de recursos computacionais requeridas por aplicações reais.

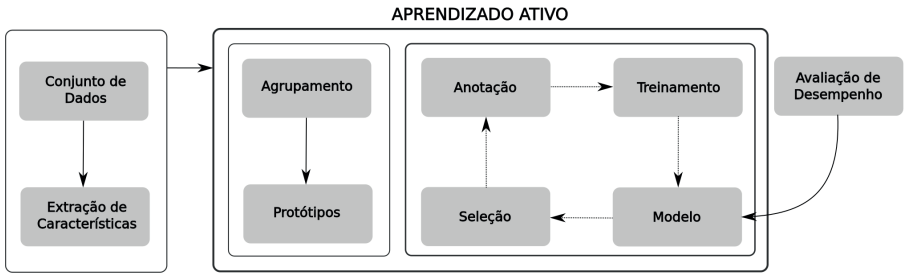


Figura 1. Contribuições para a classificação de imagens. Abordagem de aprendizado ativo e metodologia para avaliação de desempenho no aprendizado de classificadores.

$$H_{ij} = 2 \times \frac{\mu Acc_{ij} \times \mu T_{ij}}{\mu Acc_{ij} + \mu T_{ij}} \tag{1}$$

$$\Omega_i = \arg\min\{H_{ij}\} \tag{2}$$

A segunda contribuição, na área de classificação, refere-se à **inclusão de estratégias de aprendizado ativo no processo de aprendizado do classificador**, o qual a medida que aprende, seleciona e sugere rótulos para a anotação das amostras que são mais difíceis de classificação, minimizando o esforço de anotação manual de todo o conjunto de dados fornecido pelo especialista, conforme requerido no processo de aprendizado supervisionado tradicional. Uma nova estratégia de aprendizado ativo é proposta [Bressan et al. 2018b], a qual possibilita explorar melhor as amostras mais informativas para o aprendizado do classificador, por meio de critérios de seleção com base em representatividade e incerteza das amostras. Para validação da estratégia foi realizada uma avaliação extensiva, realizando comparações entre estratégias de aprendizado da literatura, utilizando diferentes técnicas de classificação e conjuntos de dados. A partir dos resultados obtidos, é possível observar que a estratégia de seleção proposta, de forma geral, atinge elevadas acurácias mais rapidamente, nas primeiras iterações de aprendizado em relação às demais estratégias comparadas.

Para a área de recuperação de imagens por conteúdo são apresentadas três principais contribuições (destacadas na Figura 2). A primeira inclui a proposta de uma estratégia para **avaliação de desempenho de descritores** [Bressan et al. 2018a], de forma a obter o par, envolvendo extrator de características e função de distância, que melhor descreve imagens médicas. Para tanto, além de extratores tradicionais, foram analisadas diferentes arquiteturas de redes neurais convolucionais para extração de características profundas. Sendo assim, foi possível avaliar o papel e o impacto das características profundas, obtidas por meio de aprendizado profundo, na recuperação de imagens médicas baseadas em conteúdo. Analisando os resultados obtidos, é possível utilizar técnicas de *transfer learning* para realizar o aprendizado a partir de um contexto geral para um contexto médico específico.

No entanto, para aproveitar ao máximo esse princípio, o processo de recuperação deve ser realizado com características gerais profundas em um determinado processo de refinamento da consulta. Para melhorar o processo de refinamento das consultas, são

apresentadas duas novas contribuições, as quais referem-se a **estratégias de aprendizado ativo para realimentação de relevância em CBIR**. As propostas [Bressan et al. 2019, Bressan et al. 2018c] têm como objetivo a obtenção de características e amostras mais informativas (similares e incertas) a cada iteração de realimentação do sistema. Sendo assim, é possível minimizar a interação do especialista e acelerar o processo de aprendizado, obtendo um classificador robusto mais rapidamente, e consequentemente, melhorando a qualidade das imagens retornadas (i.e. mais similares em relação à imagem de consulta e à expectativa do especialista).

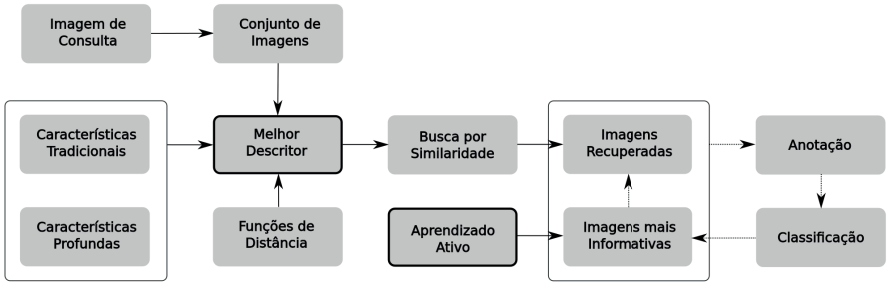


Figura 2. Contribuições para a recuperação de imagens. Abordagem de aprendizado ativo para a realimentação de relevância em CBIR.

2.1. Estratégias de Aprendizado Ativo em CBIR

Embora haja uma variedade de métodos de realimentação de relevância na literatura, a maioria deles renuncia a definição do grau de relevância e irrelevância para o usuário, descartando essa informação ao longo das iterações. No entanto, uma vez que uma consulta é refinada, a cada iteração, mais e mais imagens relevantes são retornadas, o que não contribui para o processo de aprendizado. Isso ocorre dado que é necessário analisar o melhor *trade-off* entre similaridade e diversidade, de forma a permitir que o processo escape de possíveis armadilhas locais (mínimo local).

Sendo assim, diferentemente de trabalhos da literatura prévios, é proposta a inclusão de estratégias de aprendizado ativo, a fim de obter melhorias em consultas por similaridade em sistemas CBIR. Neste caso, as imagens mais informativas a serem selecionadas são aquelas que apresentam o melhor balanceamento entre a similaridade (com a imagem de consulta) e determinados graus de diversidade e incerteza. O critério de seleção adotado deve permitir selecionar imagens de classes distintas, bem como que sejam difíceis de diferenciar.

As estratégias de aprendizado ativo propostas [Bressan et al. 2019, Bressan et al. 2018c] diferem em relação aos critérios de seleção, os quais requerem adaptações de acordo com os diferentes domínios de aplicação (por exemplo, em [Bressan et al. 2019] para diagnóstico de câncer de mama e em [Bressan et al. 2018c] para definição do vigor de sementes de soja). De forma geral, o intuito é explorar e utilizar o classificador, construído a partir das amostras mais informativas, para melhorar o processo de recuperação de imagens. Sendo assim, o modelo de aprendizado participa ativamente do processo de seleção das imagens mais informativas a serem utilizadas em seu próprio treinamento, a fim de melhorar o resultado da consulta, retornando imagens mais similares.

O conjunto de imagens selecionado é exibido para anotação (como relevante ou irrelevante de acordo com uma dada imagem de consulta) pelo especialista. A partir da primeira iteração, esse conjunto já é rotulado previamente pela instância atual do modelo. Sendo assim, o especialista precisa apenas corrigir os rótulos de imagens classificadas incorretamente. As imagens confirmadas e corrigidas pelo especialista são adicionadas ao conjunto de treinamento. Em seguida, o modelo é treinado novamente e uma nova instância é gerada. O especialista pode continuar o processo de aprendizado até que esteja satisfeito com os resultados retornados. Uma vez satisfeito, pode-se obter um modelo final de aprendizado e uma lista final ordenada por relevância (da mais similar para a menos similar) em relação à imagem de consulta.

A cada iteração do aprendizado, as imagens recuperadas devem ser aquelas que mais contribuirão para o processo de aprendizado do classificador, as quais serão obtidas a partir dos critérios de seleção adotados. Basicamente, os critérios são baseados em similaridade, diversidade e incerteza em relação à imagem de consulta. As estratégias de aprendizado ativo, em relação à similaridade, consideram tanto amostras relevantes como irrelevantes (classificadas pela instância atual do classificador). Amostras mais próximas umas das outras e que têm sido rotuladas pela instância atual como sendo de classes distintas podem ser consideradas como mais informativas (incertas). Além disso, amostras de classes distintas podem fornecer melhorias (diversidade) significativas e acelerar o aprendizado do classificador.

Para validar a abordagem de aprendizado proposta, foi realizada uma avaliação experimental extensiva utilizando diferentes domínios de aplicação. Foram realizadas comparações entre a abordagem proposta e três abordagens do estado da arte amplamente utilizadas na literatura: movimentação de centro de consulta (*Query Point Movement Strategy - QPM*), expansão de consulta (*Query Expansion - QEX*) e aprendizado ativo baseado em SVMs (SVM-AL) [Bressan et al. 2019]. Além disso, como comparativo basal para todas as abordagens mencionadas, foram também explicitados os resultados obtidos por meio do processo CBIR tradicional.

Analisando as métricas de precisão e revocação obtidas pelas abordagens, pode-se observar que a abordagem proposta, de forma geral, apresenta as melhores precisões (até 2.4 vezes) quando comparadas com as outras abordagens em análise. A abordagem proposta também apresenta um grau menor de saturação no decorrer das subsequentes iterações de realimentação em todos os conjuntos de imagens analisados, mantendo maior consistência e robustez. Além disso, é possível minimizar o tempo computacional do processo de aprendizado, uma vez que reduz (até 80%) o envolvimento do especialista. Essa redução ocorre dado que o especialista não precisa anotar (corrigir) os rótulos de todas as amostras, conforme requerido pelos trabalhos da literatura.

3. Conclusão

No projeto de mestrado foram propostas soluções que sejam capazes de lidar com aplicações reais, bem como com suas respectivas restrições (de tempo e de recursos computacionais). Os métodos foram publicados em periódicos altamente qualificados [Bressan et al. 2019, Bressan et al. 2018b] e conferências internacionais relevantes na área [Bressan et al. 2018a, Bressan et al. 2018c, Camargo et al. 2017, de Souza Junior et al. 2018].

A maior contribuição da dissertação de mestrado [Bressan 2018] consiste em incluir novas estratégias de aprendizado ativo nos processos de recuperação e de classificação de imagens. Tais estratégias foram validadas por meio de uma avaliação experimental extensiva considerando: i-) diferentes conjuntos de dados, ii-) análise dos melhores descritores (pares de extratores de características e funções de distância) tradicionais e por meio de redes neurais convolucionais; iii-) diferentes classificadores supervisionados; iv-) diferentes técnicas *baseline* e do estado da arte para comparações.

A partir dos resultados obtidos pode-se observar que as estratégias de aprendizado ativo conseguem selecionar as amostras mais informativas tanto para o processo de aprendizado do classificador como para o processo de recuperação das imagens. É possível acelerar o aprendizado de ambos os processos, minimizando a interação do especialista, obtendo acurácias de classificação elevadas e melhorando a qualidade das imagens retornadas.

Referências

Bressan, R. S. (2018). Aprendizado ativo para recuperação e classificação de imagens. Master's thesis, Universidade Tecnológica Federal do Paraná.

Bressan, R. S., Alves, D. H. A., Valerio, L. M., Bugatti, P. H., and Saito, P. T. M. (2018a). Doctor: The role of deep features in content-based mammographic image retrieval. In *IEEE 31st Intl. Symp. on Computer-Based Medical Systems (CBMS)*, pages 158–163.

Bressan, R. S., Bugatti, P. H., and Saito, P. T. M. (2019). Breast cancer diagnosis through active learning in content-based image retrieval. *Jnl. of Neurocomputing*, 357:1–10.

Bressan, R. S., Camargo, G., Bugatti, P. H., and Saito, P. T. M. (2018b). Exploring active learning based on representativeness and uncertainty for biomedical data classification. *IEEE Jnl. of Biomedical and Health Informatics*, 1:1–7.

Bressan, R. S., de Souza Junior, M., Pereira, D. F., Bugatti, P. H., and Saito, P. T. M. (2018c). Soyretrieval - técnicas de aprendizado e recuperação de imagens para análise do vigor de sementes de soja. In *Anais do XLV Seminário Integrado de Software e Hardware (SEMISH)*, pages 13–24. SBC.

Camargo, G., Bressan, R. S., Bugatti, P. H., and Saito, P. T. M. (2017). Towards an effective and efficient learning for biomedical data classification. In *IEEE 30th Intl. Symp. on Computer-Based Medical Systems (CBMS)*, pages 13–18.

de Souza Junior, M., Bressan, R. S., Pereira, D. F., Saito, P. T. M., and Bugatti, P. H. (2018). Rumo à melhoria de produtividade e sustentabilidade agrícola por meio da classificação automática do vigor de sementes de soja. In *Anais do XLV Seminário Integrado de Software e Hardware (SEMISH)*, pages 1–12. SBC.

Rao, Y., Liu, W., Fan, B., Song, J., and Yang, Y. (2018). A novel relevance feedback method for CBIR. *World Wide Web*, 21(6):1505–1522.

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

Wang, X.-Y., Yang, H.-Y., Li, Y.-W., Li, W.-Y., and Chen, J.-W. (2015). A new svm-based active feedback scheme for image retrieval. *Engineering Applications of Artificial Intelligence*, 37:43–53.

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Graduation Student Workshop Chair

PROCEEDINGS

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

Program Chair

Ticiana Linhares

Editorial

It is a pleasure to introduce the Workshop for Undergraduate Students' Work - WTAG Workshop. It is the first time such an event happens with SBBD, and we are proud to see the engagement of undergraduate students and their advisors through papers submissions.

The WTAG is an excellent opportunity to receive feedback upon ongoing undergraduate work from experienced researchers. All submitted papers received, at least, two reviews. Additionally, during the Workshop, students of selected papers have the opportunity to present their work and to receive technical and scientific comments. In this edition, we have eleven accepted papers to be published in this proceedings and presented at the Workshop.

The 2019 WTAG Workshop chair would like to thank the students and their advisors for submitting their work to the workshop. Similarly, we are very grateful to the reviewers. Their insightful comments will probably have a positive impact on the development of the different research initiatives presented in the WTAG. Finally, the WTAG Workshop chair would like to thank the SBBD 2019 organizers for their outstanding support and excellent collaboration in preparing this year's edition. We wish the community an excellent workshop and success in their future works.

Ticiana L. Coelho da Silva, UFC
Bernadette Farias Lóscio, UFPE
WTAG Program Chairs

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

SBBD Steering Commitee

Ângelo Brayner (UFC)
Bernadette Lóscio (UFPE) coordenadora da CEBD
Carina Dorneles (UFSC)
Sérgio Lifschitz (PUC-Rio)
Fábio Porto (LNCC)
Carmem Hara (UFPR)

SBBD 2019 Commitee

Steering Committee Chair
Bernadette Lóscio (UFPE)

Local Chair
José Maria da Silva Monteiro Filho (UFC, Brazil)

Full Paper Chair
Carina F. Dorneles (UFSC, Brazil)

Short Paper Chair
Fábio Porto (LNCC, Brazil)

Demos and Applications Chair
Robson L. F. Cordeiro (ICMC-USP, Brazil)

Thesis and Dissertation Workshop Chair
Jonice Oliveira (UFRJ, Brazil)

Tutorials Chair:
Altigran Soares da Silva (UFAM, Brazil)

Short course Chair
Maria Cláudia Cavalcanti (IME, Brazil)

Workshop Chair
José Antônio Macedo (UFC, Brazil)

Thesis and Dissertation Contest Chair
Caetano Traina Jr. (USP, Brazil)

Graduation Student Workshop Chair
Ticiana Linhares (UFC, Brazil)

Local Organization Committee
SBBD Local Chair: José Maria da Silva Monteiro Filho (DC/UFC)
Leonardo Oliveira Moreira (Instituto UFC Virtual/UFC)
Marum Simão Filho (UNI7)
Angelo Roncalli de Alencar Brayner (DC/UFC)
Javam de Castro Machado (DC/UFC)

Graduation Student Workshop Program Committee
Bernadette Loscio (Universidade Federal de Pernambuco)
Duncan Ruiz (Pontifícia Universidade Católica do RS)
Flávio R. C. Sousa (Universidade Federal do Ceará)
João Paulo Madeiro (Universidade Federal do Ceará)
José Monteiro (Universidade Federal do Ceará)
Karin Becker (UFRGS)
Lívia Cruz (Universidade Federal do Ceará)
Luís Gustavo Coutinho do Rêgo (Instituto Federal de Educação, Ciência e Tecnologia do Ceará)
Michele Brandão (Instituto Federal de Minas Gerais)
Nielsen Machado Ritter dos Reis (University Center - Laureate International Universities)
Regis P. Magalhães (Universidade Federal do Ceará)
Sergio Lifschitz (PUC-Rio)
Ticiana Coelho da Silva (Universidade Federal do Ceará)

Table of Contents (Graduation Student Workshop Chair)

Utilizando Informação Contextual para Refinamento de Clustering em Diversificação Visual

Lucas Almeida Silva[1,2], **Rodrigo Tripodi Calumby**[2]

[1] Universidade Federal da Bahia (UFBA)

[2]Universidade Estadual de Feira de Santana (UEFS)

lucas.almeida.las@ufba.br, rtcalumby@uefs.br

Abstract. *Traditionally, content-based image retrieval systems compute similarities using only a pair of images, disregarding the information that the spatial proximity of images in a feature space could provide. In this work, we use the information provided by the neighborhood about other images of the same collection to refine the similarity values computed, in order to improve the visual diversification in the result of the queries.*

Resumo. *Tradicionalmente, sistemas de recuperação baseados no conteúdo da imagem computam as similaridades utilizando apenas um par de imagens, desconsiderando a informação que a proximidade espacial das imagens, em um espaço de características, poderia fornecer. Neste trabalho, utilizamos da informação fornecida pela vizinhança sobre outras imagens da mesma coleção para refinar os valores de similaridade computados, afim de melhorar a diversificação visual no resultado das consultas.*

1. Introdução

A facilidade de captura de imagens e vídeos digitais, o avanço nas tecnologias de armazenamento e o crescimento acentuado de aplicações de compartilhamento na Internet têm permitido a criação e disponibilização de grandes volumes de dados. Esses dados são armazenados em mídias eletrônicas, demandando de métodos computacionais eficientes e eficazes para sua exploração. Diversos métodos têm sido propostos com o objetivo de gerar resultados melhores em variadas atividades. Dentre estas atividades, pode-se destacar: a recuperação, classificação, agrupamento e recomendação de informação.

A variedade dos tipos de dados disponíveis: textos, imagens e vídeos tem crescido. Esses tipos de dados são considerados não-estruturados e adicionam complexidade em sua análise [Jain 2010]. A recuperação de imagens baseada em texto, por exemplo, enfrenta os desafios de descrições incompletas, não precisas ou linguísticos como, ironia, sarcasmo, sinônimos ou ambiguidade, que dificultam a interpretação do interesse de busca do usuário [Pedronette et al. 2014]. Uma abordagem comumente usada para superar parte dessas limitações e desafios é a Recuperação de Imagem Baseada em Conteúdo (do inglês, *Content-Based Image Retrieval* – CBIR) [Pedronette et al. 2014]. Basicamente, um sistema de CBIR define como entrada uma imagem e, considerando propriedades visuais (e.g., cor, textura e forma) extraídas das imagens de uma base, computa as similaridades entre a imagem de entrada e as demais imagens da coleção, as ranqueando em ordem decrescente de similaridade de acordo com a função utilizada [Torres and Falcão 2006].

Apesar dos sistemas CBIR atenuarem as limitações da recuperação de imagens baseada em texto, outros desafios se apresentam, por exemplo, a subjetividade no interesse de busca. Existem consultas que não possuem apenas uma única interpretação do interesse de busca do usuário. O usuário pode ter interesses de busca diferentes, em momentos distintos, para uma mesma consulta. Afim de aumentar a cobertura de múltiplas interpretações do usuário no resultado de uma consulta, sistemas CBIR têm utilizado de técnicas de agrupamento para descobrir as possíveis interpretações, selecionando de cada grupo representantes para formar o conjunto de resposta [Calumby et al. 2017].

As funções de similaridade, tradicionalmente, computam os seus valores apenas entre dois vetores, não considerando a vizinhança espacial dos vetores no momento de definir a distância de um vetor para os outros. Trabalhos recentes na literatura propõem explorar a informação que a vizinhança de uma imagem pode fornecer sobre outras imagens em um espaço de características. Essa informação fornecida pela vizinhança é chamada de Informação Contextual (IC) e é utilizada para refinar o valor de similaridade entre duas imagens [Pedronette et al. 2014, Guimarães Pedronette and da S. Torres 2011]. Neste trabalho, é apresentada e avaliada experimentalmente uma proposta de refinamento de clustering utilizando Informação Contextual para uma coleção de imagens heterogêneas.

2. Trabalhos Correlatos

2.1. Recuperação de Imagens e Diversidade

Sistemas tradicionais de recuperação de imagem baseiam o cálculo da relevância em informações textuais associadas à imagem. Essas informações podem ser obtidas de diversas fontes, como as descrições fornecidas pelos usuários, legendas, tags, páginas web, metadados (geolocalização, data e hora, etc), palavras-chave, etc [CALUMBY 2010]. Descrever o conteúdo de uma imagem utilizando apenas das informações textuais associadas a ela pode ser impreciso, seja pela a ausência completa ou parcial de informações, ou porque elas não descrevem adequadamente o conteúdo da imagem [Pedronette et al. 2014]. Os sistemas CBIR são fundamentados no uso de descritores, que possuem dois componentes: um extrator de características e uma função de similaridade.

Além disso, a diversidade pode ser definida como um atributo de um conjunto, onde os seus itens possuem uma diferença conceitual entre si. Consultas textuais são muitas vezes ambíguas ou subespecificadas, seja porque o usuário incluiu um termo com múltiplos significados ou porque o termo especificado é insuficiente para expressar a intenção da busca. Sendo assim, os documentos recuperados para uma consulta podem transmitir informações redundantes. É verdade que o usuário pode ficar satisfeito observando apenas um aspecto no resultado. Contudo, em cenários de reais busca, a ambiguidade e a redundância podem tornar a qualidade do ranking gerado insatisfatória, pois podem privilegiar apenas um aspecto de busca em detrimento dos outros [Santos et al. 2015].

As abordagens de ranqueamento orientadas por relevância assumem que as necessidades de busca do usuário são transmitidas pela consulta e que a relevância de um documento não depende da percepção de outros documentos ou de suas relevâncias. Dado que nos sistemas CBIR as imagens são ranqueadas em ordem decrescente de similaridade, as posições iniciais do ranking podem apresentar imagens muito similares entre si, produzindo uma baixa cobertura de aspectos de busca. Sob esse enfoque, diversas abordagens

promovem a diversidade aplicando técnicas de agrupamento a fim de produzir um novo ranking que seja o máximo diverso e similar a imagem de consulta nas primeiras posições do ranking [Ji et al. 2009].

Diferente da abordagem clássica de computo das distâncias entre duas imagens, que considera apenas o par de imagens, neste trabalho o valor de distância entre duas imagens será definido a partir da contribuição da vizinhança das duas imagens, aproximando imagens mais similares e distanciando as mais dissimilares. Assim, melhorando a descoberta dos aspectos de busca quando aplicado os algoritmos de agrupamento, auxiliando na redução de aspectos redundantes.

2.2. Espaço Contextual

Considere um espaço bidimensional $\mathbb{R}^2$ construído levando em consideração pares ordenados de distância de imagens definidos por $\rho : C$ x $C \rightarrow \mathbb{R}$. Nós podemos utilizar esse espaço para analisar a similaridade da coleção de imagens para com duas imagens arbitrárias $img_i, img_j \in C$ (essas imagens são usadas como referência). A posição de uma imagem $img_l \in C$ é dada pelo par ordenado $(\rho(img_i, img_l), \rho(img_j, img_l))$, onde $\rho(img_i, img_l)$ e $\rho(img_j, img_l)$ são as distâncias de img_l para as imagens img_i e img_j, respectivamente. A informação contextual (IC) da imagem img_i é definida por suas k imagens mais próximas. Há dois métodos de re-ranqueamento baseados em Espaço Contextual (EC), os quais diferem entre si, principalmente, na função de seleção das imagens mais similares utilizadas para construir o IC, sendo eles: EC-KNN e EC-Mutual-KNN [Pedronette et al. 2014].

O algoritmo EC-KNN recebe como entrada: **Ks** - *número inicial de vizinhos no contexto*, **Ke** - *número final de vizinhos no contexto* e ***A*** - *matriz de distâncias iniciais*. O algoritmo é iterativo e executado $[K_e - K_s]$ vezes. Para cada $img_i \in C$, as distâncias para as outras imagens img_l são redefinidas utilizando a contribuição dos *k* vizinhos mais próximos (img_j). O Pseudocódigo 1 ilustra a etapa de redefinição das distâncias.

Algoritmo 1: REDEFINIÇÃO DAS DISTÂNCIAS NOS ALG. EC-KNN E EC-MUTUAL-KNN

```
para cada img_i ∈ C faça
    para cada img_l ∈ C faça
        {Considerando o espaço contextual}
        c_k ← 0; d_j ← 0
        para cada img_j ∈ selecao_vizinhos(img_i) faça
            d_j ← d_j + A[j, l] * (K − c_k)
            c_k ← c_k + 1
        fim
    fim
fim
d_i ← A[i, l]/K; d_j ← d_j/(k*(k+1)/2)
A_{t+1}[i, l] ← sqrt(d_i^2 + d_j^2)
```

O algoritmo EC-Mutual-KNN define como parâmetros de entrada: **K** - *número fixo de vizinhos*, **T** - *número de iterações*, λ - *profundidade do rank para seleção dos*

vizinhos mais próximos, ***A*** *- matriz de distâncias iniciais* e ***R*** *- lista de vetores de imagens ordenados por menor distância para cada imagem i da coleção*. As condições da formação da IC são resumidas como segue: **i)** A posição $pos(img_j, img_i) < \lambda$, onde $pos(img_j, img_i)$ retorna a posição da img_j no rank R_i da imagem img_i; e **ii)** O valor de $pos(img_j, img_i) + pos(img_i, img_j)$ é um dos *K* menores valores obtidos, considerando os ranks da imagens img_i e img_j.

3. Proposta

Neste trabalho, se propõe avaliar o impacto do uso da IC nas etapas preliminares à Diversificação Visual para cobertura de aspectos de busca em uma base de imagens heterogêneas. Os objetivos específicos desse trabalho são: A) Avaliar o impacto das combinações dos parâmetros dos algoritmos de Espaço Contextual na qualidade dos rankings; B) Avaliar o impacto das combinações dos parâmetros de execução, como: método de agrupamento, número de grupos e filtro de relevantes na qualidade do ranking gerado.

4. Metodologia

4.1. Configuração Paramétrica, Dados e Avaliação

Os Algoritmos de EC-KNN e EC-Mutual-KNN apresentados na Seção 2.2 foram adaptados refinando os valores de distâncias das matrizes de entrada A_i e retornando novas matrizes de distância A_{cs_i}. No Algoritmo de EC-Mutual-KNN os parâmetros λ e K, foram configurados como $K \leq 0.3 * \lambda$ para atender à condição do algoritmo de que $\lambda >> K$. Assim como, o parâmetro T foi definido no intervalo $T = [2, 10]$ e, para cada valor de T, o valor do parâmetro K variou entre $[2, 10]$. No algoritmo EC-KNN os parâmetros K_s e K_e foram configurados como $K_s = [2, 9]$ e $K_e = [K_s + 1, 10]$, de modo a manter o mínimo de uma iteração.

As imagens da coleção da MediaEval 2015 foram obtidas a partir de consultas por lugares turísticos realizadas no Flickr [Ionescu et al. 2015]. O resultado de cada consulta é representado por cerca de 300 imagens que são ranqueadas pelo Flickr. Além disso, na base de imagens, o resultado de cada consulta contém informações textuais das imagens recuperadas. Esta coleção de imagens possui 153 consultas, sendo 70 consultas multi-tópicos. Para a realização dos experimentos na coleção MediaEval 2015 foi selecionado um conjunto de descritores visuais e textuais. Os descritores de cor utilizados foram: ACC [Huang et al. 1997] e LUM [Lux and A.Chatzichristofis 2008]. Os descritores de textura foram Gabor [Lux and A.Chatzichristofis 2008] e PHOG [Bai et al. 2009] e o descritor textual foi BM25 [Robertson and Walker 1994]. Esses descritores foram selecionados utilizando como critério os resultados obtidos em outros trabalhos realizados [Pedronette et al. 2014, Penatti et al. 2012].

4.2. Avaliação de Eficácia da Diversificação

Os experimentos realizados serão avaliados na eficácia do aumento da diversificação, utilizando como medida central a Cluster-Recall(CR@N). A medida Cluster-Recall ($CR@N = \frac{|\bigcup_{i=1}^{N} subtopics(d_i)|}{n_s}$), onde d_i é o enésimo documento, subtopics(d_i) é o número de aspectos onde d_i é considerado relevante e n_s é o número total de aspectos para uma consulta [Zhai et al. 2003]. CR@N mensura o número de aspectos de busca recuperados

no resultado. A média harmônica de duas medidas independentes, foi utilizada para avaliar a relação Diversidade x Precisão. A medida P@N ($P@N = \frac{|d_{rel} \cap d_{rec}|}{d_{rec}}$), onde d_{rel} é o número de documentos relevantes e d_{rec} é o número de documentos recuperados. A medida F1@N é a combinação da P@N e da CR@N ($F1@N = \frac{2 * P@N * CR@N}{P@N + CR@N}$).

4.3. Características Gerais dos Experimentos

Para os experimentos, as matrizes de distâncias da coleção de imagens foram pré-computadas utilizando a função de distância euclidiana. Cada matriz de distância foi gerada sobre os vetores de características extraídos pelos descritores. As matrizes de distância possuem tamanho n x n, onde *n* é o número de vetores da base. Uma matriz de distância foi gerada para cada descritor. Para o refinamento das matrizes originais de distância foram utilizados os algoritmos de EC-KNN e EC-Mutual-KNN.

Para avaliar o impacto do ajuste das matrizes de distância no processo de diversificação, foram utilizados dois algoritmos de agrupamento: K-Medóides e Aglomerativo (Complete-link). Para cada algoritmo foi utilizado o número de grupos $k = [20; 25; 30]$. Combinado aos dois algoritmos e suas especificações de número de grupos, aplicou-se um filtro (corte) no ranking original de tamanho: top-50, top-100 e top-150, antes da execução dos algoritmos de agrupamento para sumarização e descoberta dos aspectos de busca. Nessa etapa de sumarização, a imagem com maior relevância dentro de um grupo é escolhida como seu representante para ser inserida no ranking final. Após realizada a escolha para todos os grupos, esse processo é realizado repetidas vezes, até que o ranking final esteja completo.

5. Resultados e Discussões

5.1. Avaliação do Impacto dos Parâmetros de Formação da Rank de Diversidade

Em cada combinação dos parâmetros nos algoritmos EC-KNN (K_s e K_e) e EC-Mutual-KNN (T, K e λ) foram geradas novas matrizes de distância. Entretanto, para a escolha da configuração paramétrica que oferece melhor resultado de eficácia foram gerados ranks de diversidade utilizando: (i) método de agrupamento K-medóides, (ii) n de grupos: 20 e (iii) n itens relevantes: Top-50. Os ranks foram avaliados usando P@20, CR@20 e F1@20. Depois, foi escolhida a configuração paramétrica, por descritor, que gerou melhor resultado. A Tabela 1 apresenta as melhores configurações paramétricas encontradas. Nas seções 5.2, 5.3 e 5.4 são feitas avaliações dos parâmetros: método de agrupamento, número de grupos e número de itens relevantes, variando o seus valores. Nestas avaliações, o objeto era verificar a melhora da diversificação (avaliada com CR@20) do algoritmos utilizando IC comparando-os com os resultados do algoritmos originais.

5.2. Avaliação do Impacto do Número de Grupos

Nesta avaliação, o método de agrupamento (K-Medóides) e número de itens relevantes (Top-50) foram mantidos fixos. As Figuras 1(a) e 1(b) apresentam os resultados de CR@20, que mantiveram-se estáveis com a variação do número de grupos. Apesar dos descritores de textura: Gabor e PHOG, terem obtido valores de diversidade superiores aos originais, o descritor BM25 teve resultados com contexto inferiores ao original. O aumento do número de grupos pode ter dividido em pequenos grupos um mesmo aspecto reduzindo a novidade.

Tabela 1. Melhores combinações de parâmetros dos algoritmos de Espaço Contextual KNN e Mutual-KNN

	Cluster Recall					Precisão					F1				
	EC-KNN		EC-Mutual-KNN			EC-KNN		EC-Mutual-KNN			EC-KNN		EC-Mutual-KNN		
Descritor	K_s	K_e	T	K	λ	K_s	K_e	T	K	λ	K_s	K_e	T	K	λ
ACC	5	6	9	3	10	3	4	2	3	10	2	4	2	3	10
LUM	5	8	9	4	20	6	7	9	5	20	5	8	9	4	20
GABOR	2	8	7	10	35	3	8	10	10	35	2	8	7	10	35
PHOG	2	10	10	7	30	7	10	2	7	30	2	10	10	6	20
BM25	3	4	2	4	20	8	9	4	10	35	3	4	2	4	20

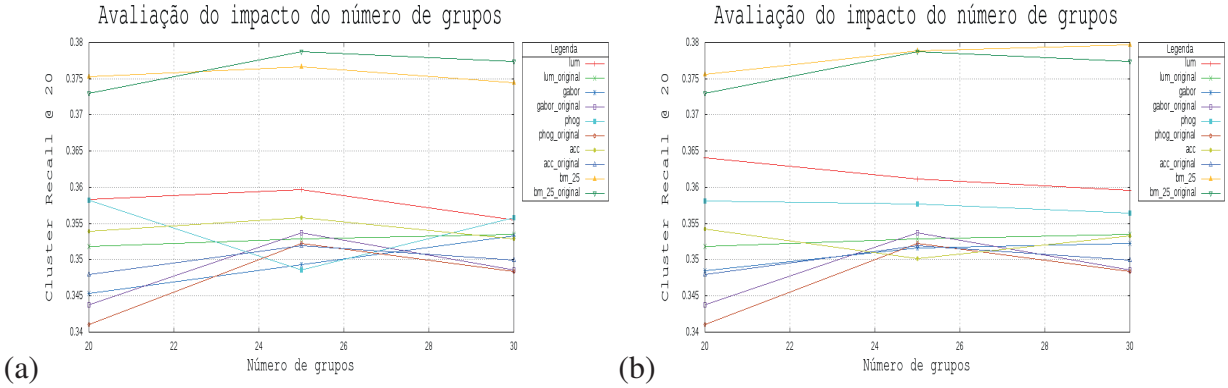


Figura 1. Avaliação do impacto do número de grupos na diversidade do rank final para os algoritmos: (a) EC-KNN e (b) EC-Mutual-KNN.

5.3. Avaliação do Impacto do Tamanho do Filtro de Itens Relevantes

Nesta avaliação, o método de agrupamento (K-Medóides) e número de grupos(20) foram mantidos fixos. Nas Figuras 2(a) e 2(b) os resultados de diversidade possuiu um crescimento quase linear à medida que o filtro aumenta, pois com um número maior de itens, há uma maior possibilidade de aumentar a cobertura dos aspectos. O descritor BM25 ganhou em diversidade $10,2\%$ (comparado ao original) quando o filtro saiu de 50 para 150 itens. O descritor ACC apresentou um crescimento de aproximadamente $9,3\%$ na diversidade. O descritor *lum* apresentou diferença de $5,5\%$, aproximadamente, comparado ao resultado original. Em ambos os algoritmos EC-KNN e EC-Mutual-KNN, a diversidade cresce com o crescimento do tamanho do filtro.

5.4. Avaliação do impacto do Algoritmo de Agrupamento

Neste avaliação, os parâmetros: número de itens relevantes (Top-50) e número de grupos (20) foram mantidos fixos. Na Figura 3 estão presentes os valores de CR@20 para cada algoritmo. Neste experimento, utilizou-se o métodos de agrupamento Kmedoids e Complete-link com os seguintes objetivos: (i) avaliar o impacto do método de agrupamento (particional ou hierárquico) na descoberta de aspectos e na formação do rank final e (ii) avaliar o impacto da utilização do EC nos resultados de cada método. Utilizamos apenas o algoritmo EC-KNN, pois possui critério de seleção de vizinhos mais simples. Em nossa avaliação, o descritor BM25 obteve menor impacto, enquanto os descritores PHOG e Gabor cresceram 5% e $2,8\%$ quando comparado com os resultados originais.

Apesar disso, a melhoria na diversidade foi pequena, mostrando que independente do método de agrupamento, o resultado de diversidade não tem variação.

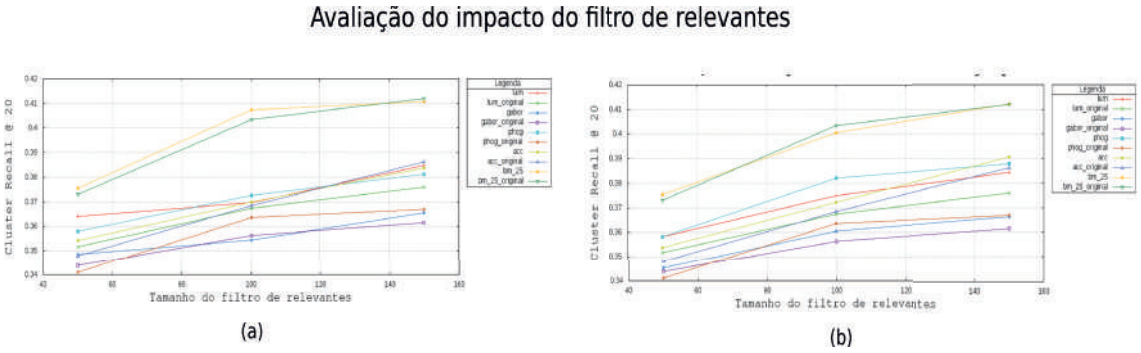


Figura 2. Avaliação do impacto do filtro de itens relevantes na diversidade do rank final para os algoritmos de: (a) EC-KNN e (b) EC-Mutual-KNN.

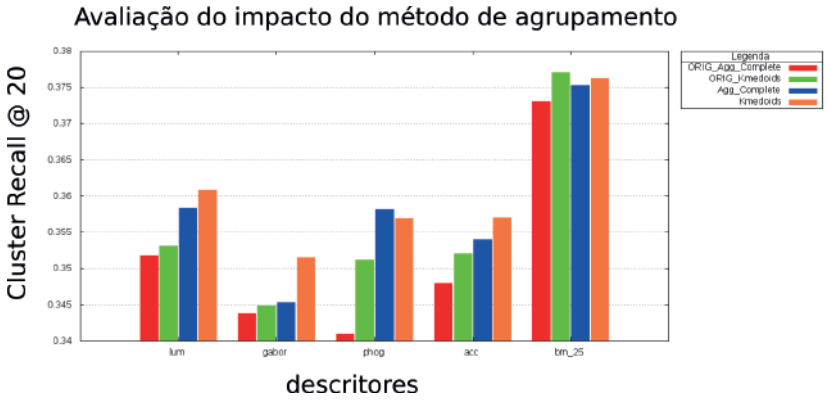


Figura 3. Avaliação do impacto do algoritmo de agrupamento na diversidade do rank final para o algoritmo EC-KNN.

6. Conclusões

Neste trabalho, foi proposto o refinamento de clustering, usando dos algoritmos de espaço contextual. Contudo, avaliamos que para matrizes de distâncias de apenas um descritor, a diferença é pequena, apesar de que para descritores visuais, obtivemos melhora. Em trabalhos futuros, pode-se realizar a combinação de matrizes de descritores, onde para a matriz resultante podemos aplicar o contexto para refinamento, afim de avaliar a sua contribuição. A análise do impacto dos parâmetros dos algoritmos de EC e da diversificação pode auxiliar na base de novos trabalhos.

Referências

[Bai et al. 2009] Bai, Y., Guo, L., Jin, L., and Huang, Q. (2009). A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. In *Proceedings of the 16th IEEE International Conference on Image Processing*, ICIP'09, pages 3269–3272. IEEE Press.

[CALUMBY 2010] CALUMBY, R. T. (2010). Recuperação multimodal de imagens com realimentação de relevância baseada em programação genética. Mestrado em ciência da computação, Universidade Estadual de Campinas, Campinas, SP.

[Calumby et al. 2017] Calumby, R. T., Gonçalves, M. A., and da Silva Torres, R. (2017). Diversity-based interactive learning meets multimodality. *Neurocomputing*, 259:159 – 175. Multimodal Media Data Understanding and Analytics.

[Guimarães Pedronette and da S. Torres 2011] Guimarães Pedronette, D. C. and da S. Torres, R. (2011). Image re-ranking and rank aggregation based on similarity of ranked lists. In Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., and Kropatsch, W., editors, *Computer Analysis of Images and Patterns*, pages 369–376, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Huang et al. 1997] Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., and Zabih, R. (1997). Image indexing using color correlograms. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pages 762–764.

[Ionescu et al. 2015] Ionescu, B., Gînsca, A.-L., Boteanu, B., Popescu, A., Lupu, M., and Müller, H. (2015). Retrieving diverse social images at mediaeval 2015: Challenge, dataset and evaluation. In *MediaEval*, volume 1436 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Jain 2010] Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666.

[Ji et al. 2009] Ji, S., Zhou, K., Liao, C., Zheng, Z., Xue, G.-R., Chapelle, O., Sun, G., and Zha, H. (2009). Global ranking by exploiting user clicks. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 35–42, New York, NY, USA. ACM.

[Lux and A.Chatzichristofis 2008] Lux, M. and A.Chatzichristofis, S. (2008). Lire: lucene image retrieval: an extensible java CBIR library. In *Proceedings of the 16th International Conference on Multimedia 2008, Vancouver, British Columbia, Canada, October 26-31, 2008*, pages 1085–1088.

[Pedronette et al. 2014] Pedronette, D. C. G., da Silva Torres, R., and Calumby, R. T. C. (2014). Using contextual spaces for image re-ranking and rank aggregation. *Multimedia Tools and Applications*, 69(3):689–716.

[Penatti et al. 2012] Penatti, O. A. B., Valle, E., and Torres, R. d. S. (2012). Comparative study of global color and texture descriptors for web image retrieval. *J. Vis. Comun. Image Represent.*, 23(2):359–380.

[Robertson and Walker 1994] Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA. Springer-Verlag New York, Inc.

[Santos et al. 2015] Santos, R. L. T., Macdonald, C., and Ounis, I. (2015). Search result diversification. *Foundations Trends Information Retrieval*, 9(1):1–90.

[Torres and Falcão 2006] Torres, R. D. S. and Falcão, A. X. (2006). Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 13:161–185.

[Zhai et al. 2003] Zhai, C. X., Cohen, W. W., and Lafferty, J. (2003). Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 10–17, New York, NY, USA. ACM.

Detectando Fake News em Manchetes - Uma Comparação de Modelos de Aprendizagem de Máquina

Manuel E. B. Filho[1], **Andreza F. de Oliveira**[1], **César L. C. Mattos**[1]

[1]Departamento de Computação (DC) – Universidade Federal do Ceará (UFC)
CEP 60440-900 – Fortaleza – CE – Brazil

{edvarfilho, andrezafernandes}@alu.ufc.br, cesarlincoln@dc.ufc.br

Abstract. *This paper aims to present a procedure to train models to classify news as true or false ('fake') from its headlines. The used dataset, available on Kaggle, is made up of news from different sources on different topics. After a methodology for pre-processing the data, 6 traditional machine learning models are used and compared in the task of classifying the veracity of the news only from the words in its headline. Finally, the obtained results are analyzed, where SVM using Gaussian kernel function was the best performing model, and further investigations are proposed.*

Resumo. *O presente artigo visa apresentar um procedimento de treinamento de modelos para classificar notícias como verdadeiras ou falsas a partir de suas manchetes. O conjunto de dados utilizado, disponível no Kaggle, é composto de notícias de diferentes fontes sobre diferentes temas. Após uma metodologia de pré-processamento dos dados, 6 modelos tradicionais de aprendizagem de máquina são usados e comparados na tarefa de classificação da veracidade da notícia apenas a partir das palavras em sua manchete. Por fim, os resultados obtidos são analisados, onde o SVM usando função de Kernel Gaussiana foi o modelo que obteve melhor performance, e novas investigações são propostas.*

1. Introdução

O atual cenário político-social mundial tem se mostrado fortemente impactado pela disseminação de *fake news*[1], as quais vem ganhando bastante notoriedade com o crescente uso de redes sociais. O papel desse meio é ressaltado por exemplo em [Delmazo and Valente 2018], onde os autores afirmam que a desinformação torna-se *fake news* em virtude de seu alcance.

Um exemplo claro do impacto da disseminação de *fake news* foi o processo eleitoral ocorrido em 2016 nos Estados Unidos. Segundo [Allcott and Gentzkow 2017], o impulsionamento de conteúdo falso teve "importância pivotal" no resultado da campanha que resultou na vitória do candidato Donald Trump. No cenário brasileiro, [Junior 2019] argumenta que a disseminação de boatos foi decisiva na percepção dos eleitores com relação aos candidatos que disputaram o processo eleitoral de 2018. Além do Brasil, países como Índia e México tem usado aplicativos de comunicação, sobretudo o WhatsApp, como principal ferramenta para a disseminação de informações falsas em geral [Melo et al. 2019].

[1]Neste trabalho considera-se *fake news* como toda notícia ilegítima e má-intencionada sendo usualmente elaborada com o intuito de disputar narrativas no debate público [Ruediger et al. 2017].

O trabalho de [Friedman 1993] afirma que para criar uma história convincente, mentirosos usam técnicas de linguagem específicas na tentativa de enganar o receptor da mensagem. Como essas formas de expressão são intrínsecas de quando contamos histórias que sabemos não ser verdadeiras, a tarefa de detecção de tais estratégias torna-se passível de ser analisada via métodos de aprendizagem de máquina.

Publicações subsequentes à de Friedman indicam que podemos aprender muito sobre as emoções e motivações de alguém ao analisar as palavras usadas na comunicação. Métodos tão simples quanto contar a quantidade de palavras usadas podem ajudar a julgar se o fato comunicado é ou não verdadeiro. O uso de determinados tipos de palavras também é um bom indicador para modelos classificadores de mentiras. Uma das hipóteses deste trabalho é que tais características típicas de inverdades possam ser automaticamente detectadas por algoritmos de aprendizagem dada uma base de dados previamente rotulada.

Visto que as *fakes news* acabam servindo como um meio de alienar a população diante de temas importantes ao debate público, o presente artigo possui o objetivo de elaborar uma metodologia capaz de classificar se determinada notícia é verdadeira ou falsa somente a partir de sua manchete, através de técnicas de pré-processamento e um comparativo de modelos gerados através de seis técnicas clássicas de aprendizagem de máquina.

No decorrer do artigo, será feita a análise de trabalhos relacionados, tal como a descrição da forma metodológica adotada para a avaliação dos seis modelos considerados. Os experimentos computacionais são feitos a partir de um conjunto de notícias reais de diferentes fontes previamente rotuladas como verdadeiras ou falsas. No final, os resultados dos diferentes modelos são discutidos e direções para investigações futuras são apontadas.

2. Trabalhos Relacionados

Pesquisas sobre detecção de notícias falsas ainda estão em um estágio inicial, já que este é um fenômeno relativamente recente, pelo menos no que diz respeito ao interesse levantado pela sociedade. Analisamos três (3) trabalhos que possuem o foco similar ao objetivo aqui apresentado:

Em [Monteiro et al. 2018] foi desenvolvida uma plataforma com o objetivo de detectar notícias falsas com o uso de uma abordagem elaborada a partir de modelos de Máquinas de Vetores de Suporte (SVM) lineares. A ferramenta, que integra o projeto *Detecção Automática de Notícias Falsas para o Português*, é treinada a partir de um conjunto de dados com notícias brasileiras no período de janeiro de 2016 a janeiro de 2018, contendo 7.200 notícias, sendo metade formada por notícias verdadeiras e metade formada por notícias falsas. Para avaliar o algoritmo foram utilizadas as métricas de precisão (*precision*), revocação (*recall*), F1-score e a acurácia. No artigo em questão é mencionado que um dos aspectos relevantes para a identificação de notícias falsas é a quantidade de erros ortográficos e advérbios, além de outros parâmetros como o número médio de substantivos, adjetivos e pronomes nos textos.

Em [Ahmed et al. 2017] é feita a descrição da construção de um modelo que detecta *fake news* a partir da análise de n-gramas. Foram usadas técnicas tradicionais como *K-Nearest Neighbors* (KNN), SVM, Regressão Logística e Árvore de Decisão. Os autores usam técnicas de limpeza e pré-processamento de dados que também serão usados no

presente trabalho, como remoção de *stop words*, técnica de *stemming* e extração de atributos, neste caso, utilizando as métricas *Term Frequency* (TF) e *Term Frequency-Inverted Document Frequency* (TF-IDF). A avaliação com melhor desempenho foi obtida usando com o uso de TF-IDF e SVM linear, sendo obtido uma precisão de 0.92.

Em [Roy et al. 2018] há a elaboração de modelos, através das técnicas CNN, Bi-LSTM e a combinação de ambas, que buscam classificar notícias em diferentes categorias: *true*, *mostly-true*, *half-true*,*barely-true*, *false* e *pants-fire*. O dataset utilizado possui seis (6) classes e cerca de 12.800 amostras. As métricas utilizadas para avaliar os modelos obtidos foram precisão (*precision*), revocação (*recall*) e F1-Score.

Com base nas análises feitas anteriormente, o nosso trabalho busca aderir práticas presentes nos trabalhos citados, desenvolvendo, assim, uma metodologia para classificação de *fake news*, que inicia no pré-processamento dos dados, com o uso de técnicas de redução de dimensionalidade por meio de combinação e seleção de atributos, até a comparação de uma gama maior de modelos de aprendizagem de máquina, cujos hiperparâmetros são selecionados por meio da técnica *grid search*, com a devida separação em conjuntos de treino, validação e teste, e o uso de validação cruzada por meio de *k-folds* para um valor de k igual a 5.

3. Metodologia

O problema considerado no presente artigo trata-se de uma classificação binária: uma notícia em questão pode ser rotulado como *falsa* ou *verdadeira*. O conjunto de dados utilizado foi extraído do Kaggle [Kaggle 2019], sendo fruto de um *web crawler* de diferentes fontes americanas. Ao todo, a base é formada por 4009 registros de notícias, sendo 2137 notícias falsas e 1872 notícias verdadeiras.

Os atributos originais desse conjunto de dados são: a URL da fonte da notícia; o *headline* (manchete) composto pelo título da notícia; o *body* (corpo) da notícia; e o *label* que indica a classe conhecida da notícia. Neste trabalho serão consideradas somente a manchete da notícia e sua classe. Ressalta-se que restringir a análise à manchete é relevante. Por exemplo, segundo pesquisa apresentada em [American Press Institute 2014], 6 em cada 10 norte-americanos leem somente o título da notícia.

O procedimento de pré-processamento, treinamento e avaliação dos modelos será descrito nas seções abaixo. Os códigos em Python usados para gerar os resultados dos experimentos podem ser encontrados no repositório `https://github.com/EdvarFilho/TrabalhoAprendizagemDeMaquina`.

3.1. Pré-processamento dos Dados

Como os dados em questão são textuais, tornou-se necessário o uso de técnicas de pré-processamento específicas. De início, realizou-se o processo de transformação dos textos para sua forma minúscula. Esse processo é importante para evitar a duplicação de atributos durante a etapa posterior de codificação via *bag of words*.

Em seguida, sinais de pontuação e palavras de parada são removidos. Essa fase é importante devido aos textos contarem muitas preposições, advérbios e demais palavras que não influem e/ou interferem no real sentido da frase, além de reduzir a dimensionalidade dos dados.

O próximo passo consiste em remover as dez palavras mais e menos frequentes, pois as mesmas devido sua raridade ou sua forte presença podem não influenciar na classificação das notícias. Em seguida, é feito o processo de extração do núcleo das palavras. A importância dessa fase é evitar a criação de atributos desnecessários no último passo a ser realizado no pré-processamento dos dados.

Por fim realizamos a codificação via *bag of words*, que consiste na geração de uma matriz em que suas colunas são todas as palavras que ocorrem no conjunto de dados, tendo um número de linhas igual ao conjunto de dados original. A matriz armazena valores um (1) nas colunas em que tal palavra compõe a sentença referente àquela linha, e zero (0) nas demais.

3.2. Redução de Dimensionalidade

Após o pré-processamento inicial houve um grande aumento no número de atributos do conjunto de dados, resultando em um total de 6333 inicial. Para contornar este problema foram usadas duas técnicas de redução de dimensionalidade: o *Fisher Score*, que funciona como quantifica a importância de cada atributo na separação entre as duas classes; e a técnicas de Análise de Componentes Principais (*Principal Component Analysis* - PCA), capaz de combinar linearmente os atributos em um espaço de menor dimensionalidade.

Primeiramente buscou-se 90% da variância explicada total dos dados via PCA, resultando em um novo conjunto de dados contendo 1443 atributos. A esse novo conjunto foi aplicada a técnica *Fisher Score* e removidos os 1000 atributos menos relevantes, sobrando 443 atributos disponibilizados aos modelos.

3.3. Modelos de Aprendizagem Supervisionada

Após a limpeza dos dados, separamos os conjuntos de treino e teste, com uma porcentagem de 25% dos dados para teste. Os modelos de aprendizagem supervisionada utilizados foram: Regressão Logística (RL), Análise do Discriminante Gaussiano (AGD), K-*Nearest Neighbors* (KNN), Árvore de Decisão (AD), *Random Forest* (RF) e Máquina de Vetores de Suporte (SVM). Os algoritmos em questão foram selecionados por serem técnicas tradicionais de aprendizagem de máquina amplamente usados na prática em tarefas de classificação.

Os hiperparâmetros dos modelos, quando necessário, foram selecionados via *grid search*, onde foi aplicada a função *GridSearchCV* disponibilizada pela biblioteca *scikit-learn* [Pedregosa et al. 2011] para todos os modelos, com exceção da Regressão Logística que foi atribuido valores pré-definidos. Tal função realiza a divisão do conjunto de treino em treino e validação, e realiza uma validação cruzada por meio da técnica *k-fold* usando o valor de k igual a cinco (5).

3.4. Métricas de Avaliação

As métricas utilizadas foram: acurácia, que é capaz de medir o quão próximo estão as classes reais das classes preditas pelo modelo; precisão, que verifica se, dados os exemplos classificados como verdadeiros, quais realmente são verdadeiros, dando assim uma fração de instâncias recuperadas que são relevantes; revocação, que dá a fração de instâncias relevantes que são recuperadas; *F1-Score*, que é utilizado em problemas de

classificação binária sendo uma média harmônica entre a precisão e a revocação. Neste trabalho instâncias da classe "falsa" são consideradas como aquelas relevantes.

A comparação dos resultados obtidos será feita ainda através de matrizes de confusão, ou seja, tabelas que mostram as frequências de classificação para cada classe do modelo, indicando os verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

4. Experimentos

Em nossos experimentos, realizamos uma divisão do conjunto de dados para treinamento e para teste. Para treino temos um total de 3009 (75% dos dados) registros e para teste o valor de 1002 (25% dos dados) registros, como já citado anteriormente.

Em seguida, realizamos o procedimento *grid search*, fazendo uso da função *GridSearchCV*, citada anteriormente, que também realiza uma validação cruzada por meio da técnica *k-fold* usando o valor de *k* igual a 5 e a realiza a divisão dos dados em validação e treino, para escolher os melhores hiperparâmetros para os modelos propostos, com exceção da Regressão Logística, no qual pré-definimos o valor do número de épocas como 1000 e o passo de aprendizagem sendo 0.0001, e a Análise de Discriminante Gaussiano que não possui hiperparâmetros. Para cada um dos algoritmos de aprendizagem de máquina de classificação temos os hiperparâmetros testados e os melhores resultados obtidos após a execução do *grid search* estarão presentes na abaixo.

- **KNN:** Hiperparâmetro: K.
  - **Valores testados:** Valor de K: 3 a 11, variando em duas unidade.
  - **Valor escolhido:** Valor de K: 3.
- **Árvore de Decisão:** Hiperparâmetros: Profundidade máxima da árvore e índice de pureza.
  - **Valores testados:** Profundidade máxima: 1 a 50, variando em apenas uma unidade. Índice de pureza: entropia ou índice de Gini.
  - **Valores escolhidos:** Profundidade máxima: 46. Índice de pureza: entropia.
- **SVM:** Hiperparâmetros: Função de kernel, valor de margem (C), gama dependendo da função de kernel, assim como o grau do polinômio.
  - **Valores testados:** Função de kernel: Gaussiano, polinomial, linear. Valor de C: 2 com potência de -5 a 16 variando-as em duas unidade. Valor de gama para kernel gaussiano: 2 com potência de -15 a 4 variando em duas unidades. Grau do polinômio para kernel polinomial: 2 a 6, variando em uma unidade.
  - **Valores escolhidos:** Função de kernel: Gaussiano. Valor de C: 8. Gama: 0.125.
- ***Random Forest*:** Hiperparâmetros: Número de estimadores e profundidade máxima para as árvores.
  - **Valores testados:** Número de estimadores: 100 a 150 , variando em uma unidade. Profundidade máxima: 1 a 10, variando em apenas uma unidade. Índice de pureza: entropia ou índice de Gini.
  - **Valores escolhidos:** Número de estimadores: 144. Profundidade máxima: 9. Índice de pureza: entropia.

5. Resultados

Após o treinamento com os melhores hiperparâmetros selecionados anteriormente, foi realizado os testes e os resultados obtidos estão apresentados na Tabela 1. Pode-se perceber que, com relação a todas as métricas avaliadas, o modelo com melhor desempenho foi o SVM, mesmo sendo suscetível a erros ocasionados por *outliers* e exemplos erroneamente rotulados. Esta afirmação sobre o SVM é embasada na análise de todas as métricas apresentadas na Tabela 1, em que temos a acurácia que indica a porcentagem das notícias classificadas corretamente em relação ao conjunto de dados, já a precisão mede a capacidade do modelo de evitar que notícias verdadeiras sejam classificadas como falsas, em relação a revocação, esta indica a porcentagem das notícias que são falsas e foram classificadas como tal assim como as verdadeiras foram classificadas corretamente. E por fim, a métrica *F1-Score* indica a relação harmônica entre a precisão e a revocação, caso o seu valor seja alto indica que a acurácia é relevante, ou seja as classificações corretas e incorretas não apresentam grande distorções, ou seja, é uma medida de confiabilidade da acurácia.

Tabela 1. Sumário dos resultados de avaliação dos algoritmos de aprendizagem na tarefa de classificação de notícias em verdadeiras ou falsas.

Modelo	**Acurácia**	**Precisão**	**Revocação**	**F1-Score**
Regressão Logística	0.52	0.26	0.50	0.35
Análise de Discriminante Gaussiano	0.82	0.84	0.83	0.83
KNN	0.66	0.69	0.65	0.63
Árvore de Decisão	0.77	0.75	0.75	0.75
Random Forest	0.81	0.82	0.82	0.81
SVM	**0.87**	**0.88**	**0.88**	**0.88**

A Regressão Logística, que é uma técnica linear, apresentou os piores resultados entre as abordagens avaliadas, indicando que os dados usados são não-linearmente separáveis. Por esse fato, a função de kernel escolhida no *grid search* para o SVM foi a Gaussiana, que permite o mapeamento por meio de uma função não-linear dos dados para um plano que se torne linearmente separável uma classe da outra. Visando uma melhor visualização das diferenças de desempenho entre os modelos, pode-se analisar os resultados obtidos nas matrizes de confusão de ambos os modelos presentes na Tabela 2. Nos dados apresentados por esta Tabela os verdadeiros positivos refere-se às *fake news* corretamente detectadas, verdadeiros negativos refere-se às notícias verdadeiras corretamente classificadas, falsos positivos refere-se às notícias verdadeiras incorretamente classificadas como *fake news*, e falsos negativos refere-se às *fake news* incorretamente classificadas como verdadeiras. É perceptível que mesmo a função custo da Regressão Logística tendo convergido, o modelo prediz que todos os dados de teste são notícias falsas.

Com relação ao resultado do KNN, é possível dizer que qualquer variância em uma palavra do título da notícia pode movê-la para uma região de notícias de outra classe diferente da sua. Isso ocorre principalmente quando a notícia é falsa, pois os núcleos de todas as palavras foram removidos, não permitindo o reconhecimento de escritas erradas, característica particular de notícias falsas. Portanto, nota-se na Tabela 2 que muitas notícias verdadeiras foram preditas como falsas.

A Análise de Discriminante Gaussiano apresentou o segundo melhor resultado, em todas as métricas avaliadas, tendo como vantagem em relação ao SVM o não uso de hiperparâmetros. O resultado positivo está apresentado na Tabela 2, onde é possível notar que a grande maioria dos dados de teste foram preditos corretamente.

Sobre a Árvore de Decisão, possivelmente a profundidade da árvore pode ter permitido um bom resultado. Caso a árvore possuísse uma profundidade menor, não seria possível separar de forma eficiente as notícias em falsas e verdadeiras. Na Tabela 2 é possível notar erros na taxa de notícias falsas que foram preditas como verdadeiras, possivelmente pelo fato de que em algum momento um dado atributo não foi tratado ou até mesmo algum registro que ficou no limiar da classificação e não foi positivamente classificado.

Tabela 2. Resultado das matrizes de confusão obtidas pelos modelos avaliados.

	RL	**K-NN**	**AGD**	**AD**	**SVM**	***Random Forest***
Verdadeiros Positivos	530	470	400	430	**460**	400
Verdadeiros Negativos	0	190	440	330	**420**	420
Falsos Positivos	0	59	130	100	**68**	130
Falsos Negativos	472	280	25	150	**50**	56

No que diz respeito ao *Random Forest*, que é uma combinação de diversas árvores de decisão via algoritmo *Bagging* (*Bootstrap Aggregating*) e subamostragem aleatória de atributos, o resultado positivo foi garantido principalmente pelo número de estimadores e pelo índice de pureza utilizado. O resultado positivo pode ser percebido na análise da matriz de confusão deste modelo na Tabela 2.

Por fim, ao analisarmos o resultado do SVM, temos alguns exemplos de classificações corretas e errôneas. Como verdadeiro positivo a notícia "*World Cup 2018: Arjen Robben retires from Netherlands duty after Sweden defeat*". Como verdadeiro negativo a seguinte manchete foi classificada corretamente: "*10 Photos Taken In The Woods That Will Give You Nightmares*". Para os casos que não foram classificados corretamente, temos "*Documenting Sports With Tech, or It Didn't Happen*", que é uma notícia verdadeira classificada como falsa, e "*Journalist Travels To North Korea, Brings Back Photos Very Different From MSM*", uma notícia falsa dita como verdadeira pelo preditor. Com isso é possível observar que algumas notícias realmente são difíceis de serem classificadas, até mesmo por humanos, devido suas semelhanças com a classe que não pertence.

6. Conclusão

Os resultados experimentais realizados indicaram que o melhor modelo obtido foi o SVM com função de kernel Gaussiana, hiperparâmetro regularizador C igual a 8 e o gama igual a 0.125. O trabalho atendeu o objetivo de avaliar diversos modelos para classificação de notícias falsas. Ressalta-se que o que dificulta a classificação de algumas notícias é a semelhança das notícias falsas e verdadeiras, reforçada pelo avanço nas técnicas de fontes ilegítimas de informação.

Investigações futuras envolvem a avaliação de novos algoritmos, como redes neurais artificias; desenvolver metodologias específicas para notícias brasileiras, fazendo uso

de bases de dados em português; explorar outras formas de extração de atributos dos dados, como descrito por exemplo em [Ahmed et al. 2017] com o uso de n-gramas.

Referências

Ahmed, H., Traore, I., and Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. In *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pages 127–138. Springer.

Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36.

American Press Institute (2014). How americans get their news. `https://www.americanpressinstitute.org/publications/reports/survey-research/how-americans-get-news/`. Acessado em 06/06/2019.

Delmazo, C. and Valente, J. C. (2018). Fake news nas redes sociais online: propagação e reações à desinformação em busca de cliques. *Media & Jornalismo*, 18(32):155–169.

Friedman, H. S., T. J. S. T.-K. C. S. J. E. W. D. L. . C. M. H. (1993). Does childhood personality predict longevity? journal of personality and social psychology. *Journal of Neuroscience*, 65(1):176–185.

Junior, G. C. (2019). Pós-verdade: a nova guerra contra os fatos em tempos de fake news. *ETD-Educação Temática Digital*, 21(1):278–284.

Kaggle (2019). Fake news detection. `https://www.kaggle.com/jruvika/fake-news-detection`. Acessado em 06/06/2019.

Melo, P., Messias, J., Resende, G., Garimella, K., Almeida, J., and Benevenuto, F. (2019). Whatsapp monitor: A fact-checking system for whatsapp. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 676–677.

Monteiro, R. A., Santos, R. L., Pardo, T. A., de Almeida, T. A., Ruiz, E. E., and Vale, O. A. (2018). Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In *International Conference on Computational Processing of the Portuguese Language*, pages 324–334. Springer.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Roy, A., Basak, K., Ekbal, A., and Bhattacharyya, P. (2018). A deep ensemble framework for fake news detection and classification.

Ruediger, M. A., Grassi, A., Freitas, A., Contarato, A., Taboada, C., Carvalho, D., Ferreira, H., Silva, L. R. d., Lenhard, P., Bastos, R., et al. (2017). Robôs, redes sociais e política no brasil: estudo sobre interferências ilegítimas no debate público na web, riscos à democracia e processo eleitoral de 2018.

OSUS: Uma visão semântica sobre os dados do Sistema Único de Saúde (SUS)

Tibet Brasil Teixeira[1], **Matheus Mayron Lima da Cruz**[1], **Vania Maria Ponte Vidal**[1]

[1]ARiDa – Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brazil

tibet.teixeira@alu.ufc.br

{matheusmayron, vaniap.vidal}@gmail.com

Abstract. *The "Sistema Único de Saúde"(SUS) is one of the largest and most complex public health systems in the world. The databases that make up this system have data that can be consumed to perform analysis and detection of problems related to population health. However, before any investigation, the data must be processed and integrated to provide a homogeneous view of the data to be used. Thus, this paper presents OSUS, a semantic view on SUS data. This work aims to semantically integrate the bases of SUS using an approach based on ontologies, to facilitate the consumption of the data.*

Resumo. *O Sistema Único de Saúde (SUS) é um dos maiores e mais complexos sistemas de saúde pública do mundo. As bases de dados que compõem esse sistema possuem dados que podem ser consumidos para realizar análises e detecção de problemas relacionados à saúde da população. Entretanto, antes que qualquer análise seja conduzida, os dados precisam ser tratados e integrados, a fim de fornecer uma visão homogênea sobre os dados a serem utilizados. Desta forma, este trabalho apresenta OSUS, uma visão semântica sobre os dados do SUS. Utilizando uma abordagem baseada em ontologias, este trabalho tem como objetivo integrar semanticamente as bases do SUS, facilitando assim o consumo desses dados.*

1. Introdução

O Sistema Único de Saúde (SUS)[1] é um dos maiores e mais complexos sistemas de saúde pública do mundo. As bases do SUS possuem informações que abrangem desde simples atendimento para avaliação da pressão arterial até o transplante de órgãos. Desta forma, uma análise aprofundada sobre os dados desse grande sistema pode auxiliar na análise e detecção de problemas existentes e na avaliação de políticas públicas.

Entretanto, antes que qualquer análise seja conduzida, é necessário resolver a heterogeneidade de dados existente entre bases de dados do SUS. Por ser um sistema grande e complexo, é bastante difícil manter o padrão entre todos os subsistemas que o compõem. Dessa forma, para que se possa alcançar a interoperabilidade entre as informações provindas entre as diferentes bases é necessário integrar semanticamente as fontes de dados.

Chamamos de integração semântica o processo que busca eliminar heterogeneidade semântica dos dados através de uma representação conceitual dos dados e de seus

[1]http://www.saude.gov.br/sistema-unico-de-saude

relacionamentos. Neste trabalho, esse processo é realizado por meio de uma abordagem que combina ontologias e tecnologias da web semântica. Esta abordagem permite que as bases do SUS sejam integradas de maneira *pay-as-you-go*. Desta forma, o processo de integração semântica das bases de dados do SUS pode ser conduzido de forma incremental, permitindo que a visão semântica - resultante da integração semântica - possa ser utilizada mesmo que ainda não esteja "completa".

O restante do artigo está organizado da seguinte forma. A Seção 2 descreve o *framework* de integração semântica utilizado na construção da visão semântica OSUS. A Seção 3 apresenta as fontes de dados integradas por este trabalho. A Seção 4 descreve a aplicação do framework de integração sobre as bases de dados do SUS e o resultado obtido. A Seção 5 apresenta alguns trabalhos relacionados, com abordagens semelhantes. Finalmente, a Seção 6 contém os trabalhos futuros e a conclusão.

2. *Framework* para Construção de uma Visão Semântica

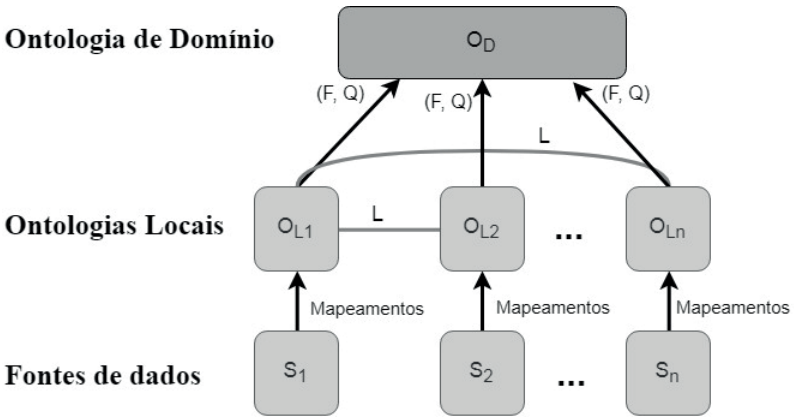


Figura 1. *Framework* de Integração Semântica

Uma visão semântica é definida como o resultado de uma integração semântica e pode ser utilizada para oferecer uma visão unificada sobre as fontes de dados por meio do uso de uma ontologia [da Cruz et al. 2019]. Na Figura 1, é mostrado um framework para integração semântica. Abaixo, é apresentado cada parte do *Framework* de Integração Semântica:

- **Ontologia de Domínio** (O_D): representação formal e explícita do conhecimento, que será compartilhada para a anotação semântica das fontes de dados;
- **Fontes de dados** ($S_1, ..., S_n$): Bases que contém os dados. Essas fontes de dados podem ser heterogêneas entre si;
- **Mapeamentos** ($M_1, ..., M_n$): Cada mapeamento M_i especifica o relacionamento entre termos da base de dados S_i e o vocabulário da ontologia local O_{Li};
- **Ontologia Local** ($O_{L1}, ..., O_{Ln}$): representação formal e explícita do conhecimento relacionado a aquela base em específico;
- **Regras de Linkage** L: conjunto de regras utilizadas para realizar a descoberta de links semânticos entre recursos das bases de dados. Esses links podem especificar, dentre outras coisas, que dois recursos representam o mesmo objeto no mundo real. Em OWL[McGuinness et al. 2004], a propriedade *owl:sameAs* é utilizada para indicar este tipo de relacionamento;

- **Regras de Fusão** F: As regras de fusão especificam como deve ocorrer a fusão dos recursos relacionados pela propriedade *owl:sameAs* em uma representação única, clara e concisa;
- **Métricas de Avaliação de Qualidade** Q: conjunto de métricas de avaliação de qualidade, que são utilizadas para quantificar a qualidade das fontes de dados, dessa forma, definindo a mais confiável entre elas.

Por meio desse *framework*, a construção da visão semântica pode ser feita de forma *pay-as-you-go* [Madhavan et al. 2007], permitindo que novas bases, ou novas informações ainda não mapeadas das bases já incluída, possam ser acrescentadas à visão de forma incremental. O processo de integração semântica de uma fonte S_k segue os seguintes passos:

1. Especificação de um conjuntos de mapeamentos M_k que mapeiam os termos utilizados pela fonte S_k em uma subconjunto de termos O_{Lk} da ontologia O_D. Caso a ontologia de domínio não contenha os conceitos existentes na fonte S_k, novos conceitos devem ser adicionados a ontologia O_D para viabilizar a inclusão de S_k;
2. Especificação das regras de *Linkage* que serão utilizadas para descoberta de links semânticos, incluindo as regras capazes de descobrir links *owl:sameAs*;
3. Especificação das regras de fusão F para as propriedades de S_k. Essas regras são definidas com base na qualidade da fonte de dados e são necessárias para resolução de conflitos gerados por inconsistência dos dados [Mendes et al. 2012].

3. As fontes de dados integradas pelo OSUS

A seguir, serão apresentadas as fontes de dados integradas pela visão semântica construída OSUS. Essas fontes foram disponibilizadas pela Governança Inteligente em Sistemas de Saúde (Gissa). Segundo a APRECE[2], o Gissa é um sistema informatizado e que o gestor poderá acompanhar em tempo real e direto no seu celular os indicadores do município, inteligência epidemiológica, inteligência normativa, inteligência administrativa, inteligência da gestão compartilhada, inteligência da gestão do conhecimento, todo o acompanhamento do equilíbrio orçamentário do município, dentre muitos outros.

Para a validar a visão semântica obtida como resultado deste trabalho, foi utilizado as bases de dados referentes ao município de Tauá, no Ceará. Vale ressaltar que a visão semântica construída não está limitada apenas a Tauá, uma vez que a estrutura de cada uma das bases de dados não é distinta entre as diferentes cidades do país.

Essas bases de dados estão em formato relacional e seus esquemas são compostos por: (a) uma tabela relacionada ao SINASC; (b) uma tabela relacionada ao SIM; (c) trinta e quatro tabelas relacionadas ao e-SUS. Na validação da visão semântica, foram utilizados dados referentes ao município de Tauá onde a base do SINASC possui 13692 registros, a base do SIM possui 5963 registros e, por fim, a base do e-SUS possui um total de 57793 registros em apenas uma de suas maiores tabelas, sendo a maior fonte de dados entre as três.

3.1. Sistema de Informações sobre Nascidos Vivos - SINASC

O SINASC é uma fonte de dados que possui informação sobre os nascimentos de todo o território brasileiro. Ele foi implantado oficialmente a partir de 1990 e sua implantação

[2] `http://tiny.cc/gissa`

ocorreu de forma lenta e gradual. De acordo com o DATASUS[3], o sistema possui vários benefícios, como subsidiar as intervenções relacionadas à saúde da mulher e da criança para todos os níveis do Sistema Único de Saúde (SUS), ações de atenção à gestante e ao recém-nascido, além do acompanhamento da evolução das séries históricas do SINASC que permite a identificação de prioridades de intervenção, o que contribui para efetiva melhoria do sistema.

Além disso, o DATASUS também cita algumas funcionalidades do sistema, que são declaração de nascimento informatizada, geração de arquivos de dados em várias extensões para análises em outros aplicativos, retroalimentação das informações ocorridas em municípios diferentes da residência do paciente, controle de distribuição das declarações de nascimento (Municipal, Regional, Estadual e Federal), transmissão de dados automatizada utilizando a ferramenta sisnet gerando a tramitação dos dados de forma ágil e segura entre os níveis municipal, estadual, federal e *backup on-line* dos níveis de instalação (Municipal, Regional e Estadual).

3.2. Sistema de Informações sobre Mortalidade - SIM

O SIM é uma fonte de dados que possui informação sobre as mortalidades no território nacional. Ele foi desenvolvido pelo Ministério da Saúde em 1975. De acordo com o DATASUS[4], o sistema possui os benefícios de produção de estatísticas de mortalidade, construção dos principais indicadores de saúde e análises estatísticas, epidemiológicas e sociodemográficas.

Além disso, o DATASUS também apresenta as seguintes funcionalidades: declaração de óbito informatizada, geração de arquivos de dados em várias extensões para análises em outros aplicativos, retroalimentação das informações ocorridas em municípios diferentes da residência do paciente, controle de distribuição das declarações de nascimento (Municipal, Regional, Estadual e Federal), transmissão de dados automatizada utilizando a ferramenta sisnet gerando a tramitação dos dados de forma ágil e segura entre os níveis municipal, estadual, federal e backup on-line dos níveis de instalação (Municipal, Regional, e Estadual).

3.3. e-SUS

De acordo com o DATASUS[5], o e-SUS é uma das estratégias do Ministério da Saúde para desenvolver, reestruturar e garantir a integração desses sistemas, de modo a permitir um registro da situação de saúde individualizado por meio do Cartão Nacional de Saúde (CNS). O nome, e-SUS, faz referência a um SUS eletrônico, cujo objetivo é sobretudo facilitar e contribuir com a organização do trabalho dos profissionais de saúde, elemento decisivo para a qualidade da atenção à saúde prestada à população.

4. Construção da Visão Semântica OSUS

Para a construção da visão semântica OSUS, foi utilizado o *framework* apresentado na seção 2. Com a finalidade de validar a visão construída e mostrar seu uso, a visão semântica materializa um mashup dos dados referentes a cidade de Tauá. A figura 2

[3] http://tiny.cc/datasus-sinasc
[4] http://tiny.cc/datasus-sim
[5] http://tiny.cc/datasus-esus

mostra os passos realizados para a construção desta visão semântica, incluindo o passo onde essa visão é utilizada na materialização do mashup de dados.

Inicialmente, foram selecionadas bases de dados que seriam semanticamente integradas. As bases escolhidas foram SIM, SINASC e e-SUS, como já foi discutido na seção anterior. Terminado o passo da seleção, após uma análise dos dados presentes em cada uma dessas bases, a ontologia de domínio[6] foi modelada. Na ontologia estão presentes conceitos como *osus:Gestação* e *osus:Obito*, importantes para representar as informações provindas das bases de dados SIM e SINASC, respectivamente. Em seguida, os esquemas de cada uma das bases foram então mapeados para os conceitos estabelecidos na ontologia. Esses mapeamentos foram feitos utilizando a linguagem R2RML, visto que as bases estão em formato relacional e a linguagem R2RML é um padrão estabelecido pela W3C para realizar este tipo de mapeamento.

No passo 4, foram definidas as regras de *linkage* para realizar a descoberta de links semânticos *owl:sameAs*. As regras de linkage foram criadas utilizando a linguagem estabelecida pelo [Volz et al. 2009]. Uma visão geral destas regras é apresentada abaixo:

- **Linkage entre SIM - SINASC**: A regra de *linkage* para realizar essa descoberta funciona da seguinte forma:
  - Sub-regra 1: Caso os indivíduos possuam o mesmo Número de Declaração de Nascido Vivo, um link é estabelecido entre eles - Esta regra é importante, principalmente, para estudos que buscam investigar problemas relacionados a natalidade infantil e neonatal;
  - Sub-regra 2: Os indivíduos que não tiveram nenhum link estabelecido pela sub-regra 1, são comparados entre si por meio de uma função de similaridade que leva em conta: nome da mãe, sexo, local de nascimento e data de nascimento. Os pares que possuem possuem maior similaridade e cuja similaridade é maior que 0.98, tem um link estabelecido entre eles.
- **Linkage entre SIM - ESUS** : O link entre SIM e ESUS é feito utilizando o Número do Cartão Nacional de Saúde e por meio de uma função de similaridade entre os nomes dos indivíduos. Desta forma, os indivíduos que têm o mesmo número CNS e possuem o nome mais similar são associados por um link *owl:sameAs*;
- **Linkage entre SINASC - ESUS** : Nenhuma regra de linkage foi estabelecida entre essas bases.

Por fim, foi estabelecido como seria realizada a fusão dos dados e as métricas de qualidade por meio da linguagem utilizada pela ferramenta SIEVE [Mendes et al. 2012]. De maneira geral, as informações relacionadas ao nascimento dos indivíduos tem mais qualidade se provindas do SINASC. Informações relacionadas ao Óbito são mais confiáveis se vindas da base SIM. E informações como nome e outras informações pessoais estão melhor preenchidas no sistema ESUS. A resolução de conflitos busca sempre manter a informação da base considerada mais confiável. Isso finaliza a construção da visão semântica OSUS.

A materialização para se obter um mashup de dados é um passo que pode ser feito para que viabilizar a consulta sobre os dados semanticamente integrados. A materialização de um mashup segue os seguintes passos: (1) Os mapeamentos são

[6] `http://tiny.cc/onto_dominio`

processados e utilizados para transformar os dados em RDF (esse passo é chamado de materialização das visões exportadas); (2) Em seguidas, a descoberta dos links owl:sameAs é feita, gerando um conjunto links que ligam recursos de diferentes bases; (3) Por último, a regras de fusão são executadas, gerando como resultado o mashup de dados baseado na visão semântica OSUS. Todo esse processo foi executado sobre os dados de Tauá.

Figura 2. Passos da integração semântica do OSUS

5. Trabalhos relacionados

Em [Paiva et al. 2016], é proposto uma integração dos dados do Portal da Transparência do Governo Federal Brasileiro utilizando um data warehouse. Os autores apresentam o processo de integração de esquemas proposto por [Batini et al. 1986] e [Mello 2002], em que os passos da integração é seguido por pré-integração, comparação de esquemas, conformação de esquemas e junção e reestruturação.

Em [Pereira et al. 2015], Lucas Pereira et al. propõem uma ontologia para integração de bases do SUS voltado ao domínio de doação de órgãos e tecidos. A ontologia proposta foi utilizada no desenvolvimento do sistema Sincap, cujo o objetivo e apoiar o processo de doação de órgãos do Centro Nacional de Captação e Doação de Órgãos (CNCDO) do estado do Espírito Santo. A ontologia foi modelada utilizando a notação OntoUML[7] e construída utilizando o modelo de objetos.

Em [Brenas et al. 2019], Jo Hael Brenas et al. propõem uma ontologia de experiências adversas na infância (ACE) para vigilância, pesquisa e avaliação em saúde mental. Para isso, foi utilizado a lógica ALCRIQ, uma sub-lógica da Web Ontology Language[8] (OWL). A ontologia apresentada foi implementada e disponibilizada à comunidade de saúde mental e ao público através do repositório BioPortal.

6. Trabalhos Futuros e Conclusão

Foi apresentado neste trabalho o OSUS, uma visão semântica sobre os dados do Sistema Único de Saúde (SUS), que visa facilitar a análise dos dados das bases do SINASC, SIM e do e-SUS, respondendo a questões que não poderiam ser resolvidas sem uma camada de integração semântica. Dessa forma, é possível obter mais informações relevantes das pessoas cadastradas e, assim, encontrar um modo de melhorar a vida de uma determinada população.

[7] https://ontouml.org
[8] http://tiny.cc/w3_owl

Como trabalhos futuros, é pretendido construir uma aplicação bastante intuitiva para que usuários de diferentes áreas, que não possuem necessariamente um conhecimento sobre Web Semântica, possam acessar os dados e realizar análises sobre eles. Além disso, serão adicionadas as outras fontes de dados que fazem parte do SUS, como por exemplo o Sistema de Informação do Programa Nacional de Imunização (SI-PNI) e o Sistema de Informação de Agravos de Notificação (SINAN), criando uma grande base de dados interligados da saúde do Brasil. E, posteriormente, publicar toda a especificação apresentada neste trabalho.

Referências

Batini, C., Lenzerini, M., and Navathe, S. B. (1986). A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.*, 18(4):323–364.

Brenas, J. H., Shin, E. K., and Shaban-Nejad, A. (2019). Adverse childhood experiences ontology for mental health surveillance, research, and evaluation: Advanced knowledge representation and semantic web techniques. *JMIR Ment Health*, 6(5):e13498.

da Cruz, M. M., Avila, C., Vidal, V. M., and Junior, N. (2019). Semanticsus: Um portal semântico baseado em ontologias e dados interligados para acesso, integração e visualização de dados do sus. In *Anais Estendidos do XIX Simpósio Brasileiro de Computação Aplicada à Saúde*, pages 13–18, Porto Alegre, RS, Brasil. SBC.

Madhavan, J., Jeffery, S. R., Cohen, S., Dong, X. L., Ko, D., Yu, C., and Halevy, A. (2007). Web-scale data integration: You can only afford to pay as you go.

McGuinness, D. L., Van Harmelen, F., et al. (2004). Owl web ontology language overview. *W3C recommendation*, 10(10):2004.

Mello, R. d. S. (2002). Uma abordagem bottom-up para a integração semântica de esquemas xml. Master's thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, BR-RS.

Mendes, P. N., Mühleisen, H., and Bizer, C. (2012). Sieve: Linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, EDBT-ICDT '12, pages 116–123, New York, NY, USA. ACM.

Paiva, E., Revoredo, K., and Baião, F. (2016). Dw-cgu: Integração dos dados do portal da transparência do governo federal brasileiro. *iSys - Revista Brasileira de Sistemas de Informação*, 9(1):6–32.

Pereira, L., Calhau, R., Sérgio, P., Santos Júnior, P., and Costa, M. (2015). Ontologia de domínio de doação de Órgãos e tecidos para apoio a integração semântica de sistemas.

Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G. (2009). Silk-a link discovery framework for the web of data. *LDOW*, 538:53.

Desenvolvimento de um aplicativo *mobile* para doação de animais de estimação

Tatiana Tozzi[1], **Daniel Fernando Anderle**[2], **Rodrigo Ramos Nogueira**[3]

[1, 2] Instituto Federal Catarinense – Campus Camboriú (IFC) – Camboriú, SC – Brasil
[3] Universidade de Coimbra – Coimbra - Portugal

tatitozzi@hotmail.com, daniel.anderle@ifc.edu.br,
rodrigonogueira@dei.uc.pt

Abstract. *This article presents the modeling of a database for a system that aims to broaden the advertising of animals that are lost, located or for adoption. Initially, a survey was made of the main current technologies that can be used to locate, identify and adopt domestic animals, through this knowledge the modeling of the system was carried out.*

Resumo. *Este artigo apresenta a modelagem de um banco de dados para um sistema que visa ampliar a divulgação de anúncios de animais que encontram-se perdidos, localizados ou para adoção. Inicialmente foi realizado um levantamento das principais tecnologias atuais que podem ser utilizadas para a localização, identificação e adoção de animais domésticos, através deste conhecimento foi realizado a modelagem do sistema.*

1. Introdução

Animais domésticos ou animais de companhia fazem parte do cotidiano dos seres humanos desde os primórdios. Atualmente, os animais domésticos são considerados como membro da família, conforme aponta o estudo realizado pela Associação Brasileira da Indústria de Produtos para Animais de Estimação (ABINPET) em 2016 [MELO, 2016]. Os animais domésticos são representados em maioria pelas espécies felina e canina. Cães e gatos são excelentes companheiros dos seres humanos, além de proporcionarem conforto, amizade, e auxiliarem no tratamento de doenças [CARVALHO, 2016].

Um animal doméstico necessita de cuidados para sua sobrevivência, saúde e bem-estar. A definição de bem-estar se relaciona com o conceito das cinco liberdades: o animal não deve passar fome ou sede, nem ter uma nutrição deficiente; ser livre de desconforto; ser livre de dor, doenças ou lesões; livre de medo e estresse; livre para expressar seu comportamento normal [GUERIN, 2009].

Em muitos casos a guarda responsável não é aplicada, e o animal sofre maus-tratos e abandono, por isso, é primordial que seja divulgado e praticado a guarda responsável. “Animais não são humanos e não são brinquedos, como bem recorda Brügger, são companhia, desenvolvem afetividade, responsabilidade e cuidado, mas não são coisas, são seres” [BRÜGGER, apud MEDEIROS, 2013, p.215]. Desta maneira, as ONGs de proteção

animal, Centro de Zoonoses e Protetores Independentes buscam ser o papel, de oferecer tratamento digno aos animais em situação de rua ou em abrigos.

Dado a problemática envolvendo os animais de rua, e visando a aplicação da guarda responsável, o presente trabalho teve como objetivo realizar a identificação de tecnologias que ONGs, protetores independentes e centros de zoonoses possam vir a utilizar na identificação e resgate de animais perdidos, bem como na divulgação de animais para adoção. Através deste levantamento foi realizada uma proposta para auxiliar no resgate, identificação e adoção de animais domésticos.

2. Metodologia

Com o intuito de atingir os objetivos propostos neste artigo, inicialmente foi realizado uma pesquisa de opinião com moradores da região da AMFRI[1]. O questionário foi desenvolvido utilizando a plataforma de formulários *Google Forms*. Um convite foi enviado por e-mail e mensagens para os grupos de protetores independentes e ONGs de proteção animal, visando realizar o levantamento de tecnologias que tais moradores já usaram para auxiliar no resgate, localização e adoção de animais domésticos.

Em sequência foi realizado o desenvolvimento do projeto do sistema, nesta fase foram identificados os autores do sistema, levantado os requisitos funcionais e não-funcionais, modelado o banco de dados e realizado a prototipação do sistema.

3. Revisão de Literatura

Tendo conhecimento que o principal objetivo deste trabalho foi realizar o levantamento de tecnologias que auxiliem animais domésticos e em seguida o desenvolvimento de um sistema que venha auxiliar a identificação, localização e adoção de animais, buscou-se identificar os principais trabalhos da literatura atual que tiveram objetivos semelhantes. O trabalho de Carpanezi et al. [2016] desenvolveu uma pesquisa experimental com o objetivo de desenvolver um aplicativo para promover a adoção de animis, os possíveis tutores poderiam conhecer o animal antes de realizar a adoção, tento assim conhecimento de seu comportamento, personalidade e saúde do animal.

Menezes Filho e Souza [2017], desenvolveram um sistema para registro e identificação de animais domésticos. Os autores buscaram criar uma base de dados que as prefeituras podem vir a usar para armazenarem as informações sobre a população de animais, com tal informação as prefeituras poderão planejar ações estratégicas, desde o controle de população, assim como no controle de zoonoses.

4. Resultados

Por meio de uma pesquisa de opinião, foram identificadas as principais tecnologias que os respondentes já utilizaram para auxiliar na identificação, localização e adoção de animais. Conforme demonstrado pela Figura 1 [A], 39% dos respondentes já utilizaram alguma tecnologia para auxiliar no resgate, localização e adoção de animais. Dos 100 respondentes da pesquisa 44 deste já utilizaram alguma tecnologia, conforme demonstrado na Figura 1 [B], a principal tecnologia utilizada são as redes sociais.

[1] Associação dos Municípios da Foz do Rio Itajaí - SC

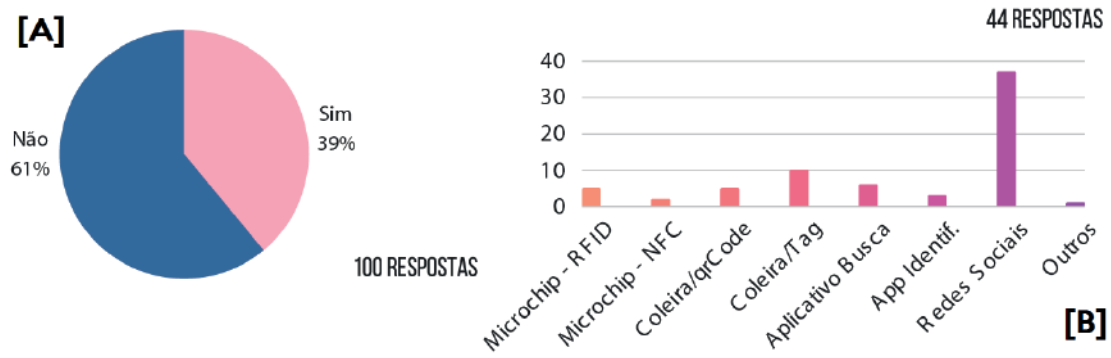


Figura 1. [A] Resultado de tutores que já utilizaram alguma tecnologia. [B] Tecnologias já utilizadas

A partir deste conhecimento foram pesquisadas tais tecnologias, as quais são apresentadas resumidamente através do Quadro 1.

Quadro 1. Lista de tecnologias

Tecnologia	Descrição
Microchip RFID (*Radio-Frequency IDentification*)	Método de identificação automática por meio de sinais de rádio, onde são recuperados e armazenados dados remotamente através de um dispositivo de *tags* RFID, tal dispositivo é implantado sobre a pele do animal;
Microchip NFC (*Near Fiel Communication*)	Tecnologia que possibilita a troca de informações e dados entre dispositivos assim como o RFID, porém, para acessar as informações no *microchip* basta possuir um *smartphone* compatível com essa tecnologia e se aproximar 10 centímetros do animal [COIMBRA, 2017], já o RFID necessita de um leitor específico para este fim;
Coleira com *qrCode* (*Quick Response Code*)	Consiste em uma coleira com uma medalha de identificação com *qrCode*, através da leitura do *qrCode* é possível acessar a página do animal, a qual contém informações de contato do tutor, telefone do médico veterinário, fotos e informações médicas;
Coleira com *tag*	Alternativa para utilização do *microchip* sem que este seja implantado no animal, a *tag* contendo os dados do animal (o código de identificação) é colocada na coleira do animal;
Aplicativos de busca	Podem ser utilizados para cadastrar informações do animal e dados de contato do tutor;
Aplicativos de identificação	Auxiliam a realizar na identificação do animal através de reconhecimento facial, utilizando a tecnologia de comparação de imagens (visão computacional e inteligência artificial).
Redes sociais	São grandes aliadas na procura e divulgação de animais. Principalmente o Facebook e o Instagram. Em fevereiro de 2018 foi criada da rede social Puppyfi, com o principal objetivo auxiliar os animais, assim auxiliando tutores a encontrarem seus animais desaparecidos.

Uma fez realizado o levantamento das tecnologias, foram identificadas as principais funções das mesmas, com o intuito de auxiliar no levantamento de requisitos do sistema, conforme identificado no Quadro 2.

Quadro 2. Funções das tecnologias identificadas

Tecnologias	Cadastro de animais	Identificação (foto)	Publicação (anúncio)	Monitoramento	Fotos dos animais	Geolocalizaçcao	Controle de acesso
Microchip – RFID	✓						
Microchip – NFC	✓						
Coleira com *qrCode*	✓	✓			✓	✓	✓
Coleira com *Tag*	✓	✓			✓		✓
Aplicativo de Busca	✓	✓	✓		✓	✓	✓
Aplicativo de Identificação	✓	✓			✓		
Redes sociais	✓		✓		✓	✓	✓
Este trabalho	✓	✓	✓	✓	✓	✓	✓

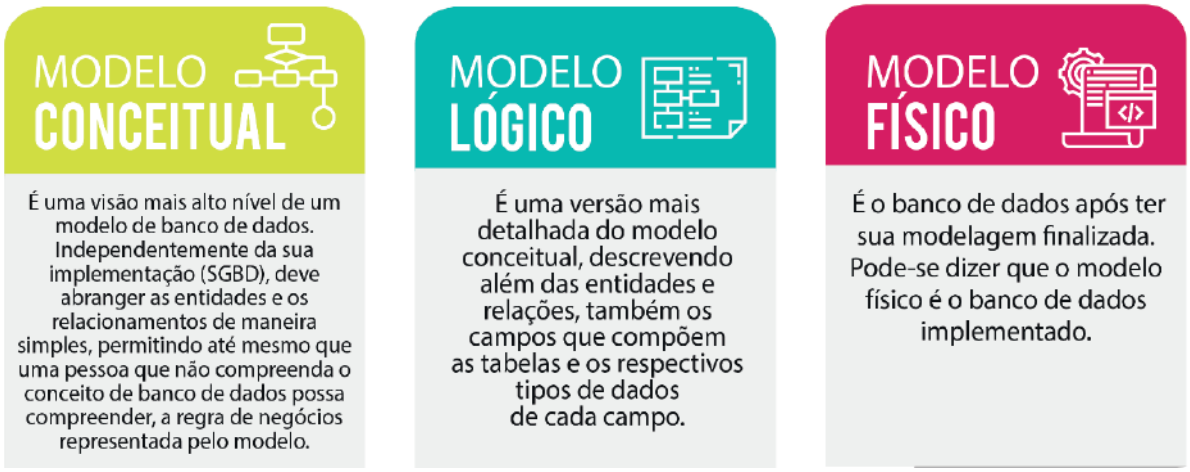
5. Discussões

O projeto de banco de dados do Sistema foi desenvolvido através da criação e modelagem conceitual, lógica e física. De acordo com Heuser [2009], um modelo de banco de dados "[...] é uma descrição dos tipos de dados e informações que estão armazenadas em um banco de dados." Para Elmasri e Navathe [2011], "um banco de dados é uma coleção de dados relacionados, sendo de fatos conhecidos e que podem ser registrados e os quais possuem algum significado implícito".

A modelagem do banco de dados auxilia na organização do pensamento dos sobre os dados, auxiliando na demonstração do significado e aplicação dos mesmos, e proporcionado estabelecer o vínculo das necessidades dos usuários são atendidas pelo sistema. Além de reduzir a complexidade do projeto facilitando a compreensão e a manipulação dos dados [MULLER, 2002].

A modelagem de dados é dividida em três modelos [NOGUEIRA, 2016], conforme ilustrado pela Figura 2.

Figura 2. Tipos de modelagem de dados



5.1. Modelo conceitual

O modelo conceitual descreve a estrutura do banco de dados independente da implementação em um SGBD (Sistema de gerenciamento de banco de dados). A técnica mais utilizada de modelo conceitual é a entidade-relacionamento (ER), sendo representado por um diagrama, chamado diagrama entidade-relacionamento [HEUSER, 2009].

Neste diagrama são organizadas as entidades do banco de dados, suas relações, especificações, atributos e cardinalidade. A Figura 3 apresenta o modelo conceitual desenvolvido para o sistema proposto neste trabalho.

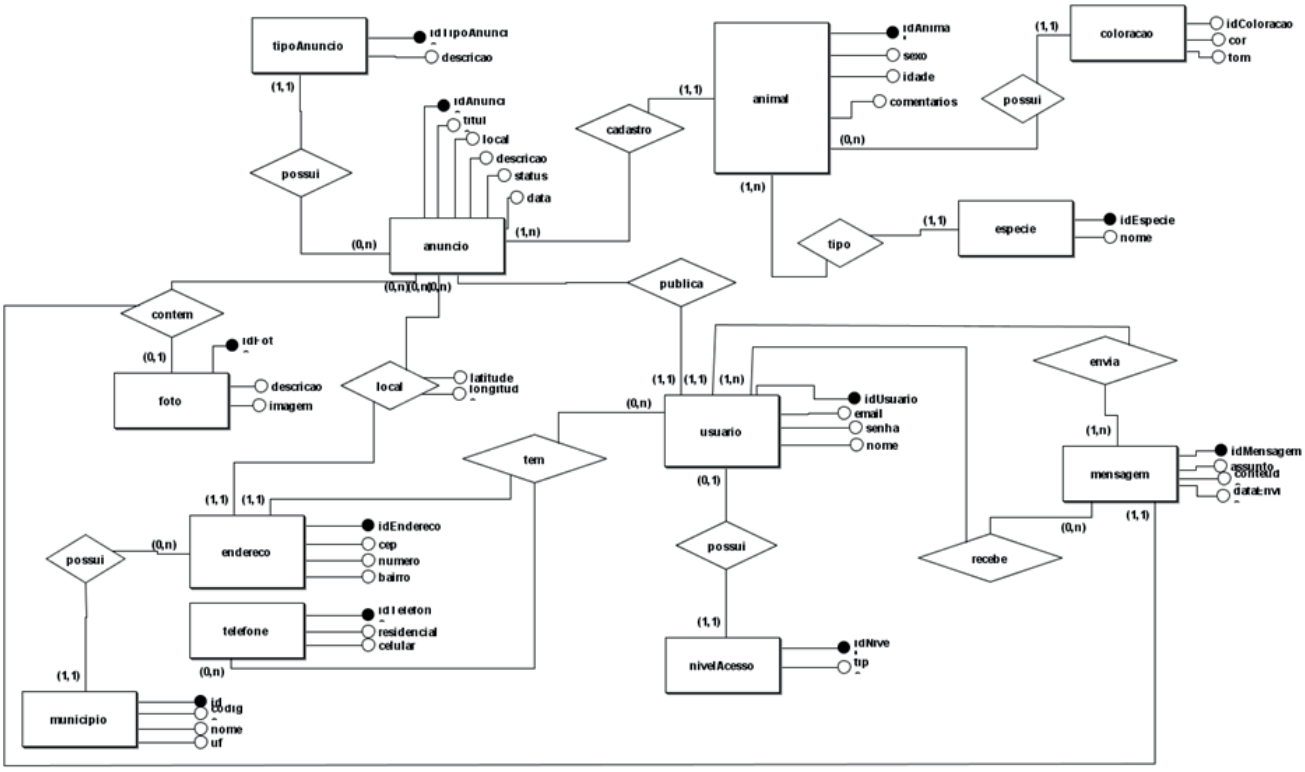


Figura 3. Modelo Conceitual

5.2. Modelo Lógico

O modelo lógico é semelhante ao modelo conceitual, o modelo lógico também possui a representação de todos os objetos e seus respectivos relacionamentos. Sendo que este modelo representa a estrutura dos dados do banco de dados vista pelo usuário do SGBD [HEUSER, 2009]. O diagrama mostra as ligações entre as tabelas do banco de dados, são representadas as chaves primárias e estrangeiras (relacionamentos) e a estrutura das tabelas do banco de dados.

Segundo Heuser [2009], "um modelo lógico de um banco de dados relacional deve definir quais as tabelas que o banco de dados contém e, para tabela, quais os nomes das colunas". Através desse modelo é definido como o banco de dados será implementado em um SGBD. Na Figura 4 é apresentado o modelo lógico, sendo que se refere a um SGBD relacional, onde os dados são organizados na forma de tabelas.

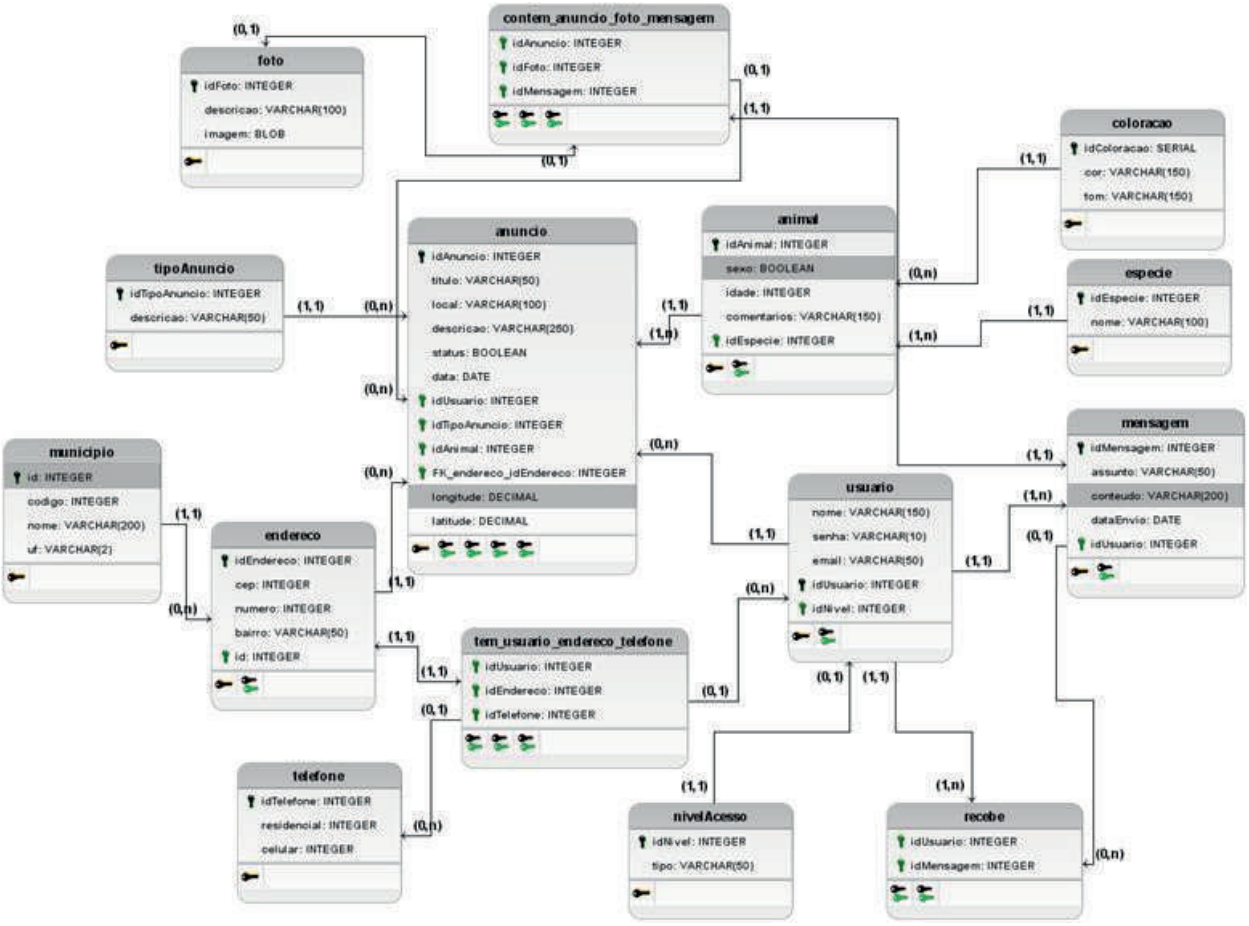


Figura 4. Modelo Lógico

5.3. Modelo Físico

Para Machado e Abreu [2012] o modelo físico do banco de dados, "[...] detalha o estudo dos métodos de acesso do SGBD para elaboração dos índices de cada informação colocada nos modelos conceitual e lógico". O modelo físico é gerado a partir do modelo lógico, tendo como resultado um script, sendo criado como base a semântica do SGBD, contendo a conotação e os tipos de dados suportados pelo SGBD. Um exemplo do modelo físico é demostrado através do quadro 3.

5.4. Dicionário de dados

O dicionário de dados serve como depósito central de dados para todos os envolvidos do desenvolvimento de um sistema. No dicionário de dados são definidos e representados dos elementos dos dados que são armazenados no banco de dados, a criação de um dicionário de dados é necessária, pois, ao ampliarmos os detalhes dos objetos criados no modelo conceitual, muita das explicações necessárias para se entender o funcionamento do banco de dados é de difícil inclusão no diagrama.

A criação do dicionário de dados possibilita realizar a descrição de muitas informações usadas na construção do sistema e é considerado um dos documentos mais importantes na documentação de um sistema. Os Quadros de 4 a 7 apresentam as principais tabelas do sistema que foram representadas no dicionário de dados. Sendo que Tabela é o nome da tabela definida no banco de dados.

A Coluna 1 são os atributos, ou seja, as informações que vão ser guardadas em cada tabela, o Tipo define o valor da coluna e a quantidade de caracteres para o armazenamento do seu conteúdo, PK (*Primary Key*) identifica se a coluna é uma chave primária, já em FK (*Foreign Key*) identifica se a coluna é uma chave estrangeira, e qual tabela e coluna possui um relacionamento e por último em Descrição serve para descrever o que é a coluna ou para informações adicionais.

Quadro 4. Dicionário tabela usuários

Coluna	Tipo	PK	FK (tabela/coluna)	Descrição
idUsuario	serial	PK		Atributo identificador da tabela usuário
nome	varchar (150)			Nome do usuário
email	varchar (50)			E-mail do usuário
senha	varchar (10)			Senha do usuário
idNivel	integer		nivelAcesso/idNivel	Referente ao nível de acesso do usuário

6. Considerações finais

Este trabalho apresenta o levantamento de tecnologias que visam auxiliar na identificação, localização e adoção de animais domésticos. Em sequência a tal levantamento foi realizado a criação da modelagem do banco de dados de um sistema que visa promover a ampliação da divulgação de animais domésticos.

Através deste trabalho foi possível conhecer a opinião dos respondentes da pesquisa de opinião e seu conhecimento e uso de tecnologias que auxiliam os animais domésticos. A partir deste conhecimento e da identificação das tecnologias, visa-se por meio das funções dessas tecnologias criar um sistema que agiliza e amplie a divulgação de anúncios de animais, assim tornando esses anúncios mais visualizados e melhorando as chances de adoção dos animais.

Referências

Carpanezi, Caroline Aparecida; Tomazela, Maria das Graças J. M.; Pontes, Aldo. (2016) "Desenvolvimento de um aplicativo mobile para adoção de animais de estimação", www.fatecid.com.br/reverte/index.php/revista/article/view/183, February.

Carvalho, Luciana. (2016) “9 benefícios que bichos de estimação trazem à saúde”, In: EXAME, https://exame.abril.com.br/estilo-de-vida/9-beneficios-que-bichos-de-estimacao-trazem-a-saude, September.

Coimbra, Diego da Silva. (2016) “O Uso da Tecnologia NFC na Identificação PET”, http://www2.uesb.br/computacao/wp-content/uploads/2014/09/O-Uso-da-Tecnologia-NFC-da-Identifica%C3%A7%C3%A3o-PET-DIEGO-COIMBRA.pdf, February.

Elmasri, R.; Navathe, S. B. (2011) “Sistemas de Banco de Dados”, Pearson Addison Wesley, 6. ed.

Guerin, K. (2009) “Programa permanente de controle reprodutivo de cães e gatos no Município de São Paulo”, In: Programa Permanente de Controle Reprodutivo de Cães e Gatos Relacionando o Impacto na Sociedade.

Heuser, Carlos Alberto. (2009) “Projeto de banco de dados”, Bookman, 6. ed.

Machado, Felipe Nery Rodrigues; Abreu, Maurício Pereira de. (2012) “Projeto de banco de dados: uma visão prática”, Érica, 17. ed.

Medeiros, Fernanda Luiza Fontoura de. (2013) “*Direito dos animais”*, Livraria do Advogado.

Melo, Luísa. (2016) “Como o brasileiro cuida e quanto gasta com seus pets”, In: EXAME, https://exame.abril.com.br/negocios/como-o-brasileiro-cuida-e-quanto-gasta-com-seus-pets, October.

Menezes Filho, Carlyle Torres B. de.; Souza, José de Lima de. (2017) “Registro geral de Animais (RGA): um sistema para o registro e identificação de animais de companhia”, https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/176967/RGA.pdf, April.

Nogueira, Rodrigo Ramos. (2016) “Passo a passo para realizar a modelagem de dados”, In: Revista SQL magazine. Ed. 138, p. 6.

Classificação de Fake News com Textos de Notícias em Língua Portuguesa Integrando Data Warehousing e Machine Learning

Roger Oliveira Monteiro[1], Rodrigo Ramos Nogueira[2], Stefano Soares[2], Daniel Anderle[2]

[1]Centro Universitário Leonardo da Vinci (UNIASSELVI)
Rodovia BR-470, Km 71, 1.040 - Benedito, Indaial - SC, 89130-000

[2]Instituto Federal Catarinense (IFC)
R. Joaquim Garcia, s/n - Centro, Camboriú - SC, 88340-055

{roger.o.monteiro,danielfernandoanderle}@gmail.com, rodrigo.nogueira@ifc.edu.br, stefano.xavier@hotmail.com

Abstract. *With the rapid advancement of technology, and the easy access and dissemination of information, the term fake news has gained worrisome attention. Thus, the purpose of this paper is to use machine learning methods to discover, classify and store fake news texts for later application to ETL of a Data Warehouse and a query environment, that will contribute to future research. For this, a dataset was created and the Logistic Regression, Naive Bayes and SVM methods were evaluated. Finally, the work has the selection of the best method, which was inserted in an online evaluation system.*

Resumo. *Com o rápido avanço da tecnologia, e o fácil acesso e disseminação de informações, o termo fake news vem ganhando preocupante atenção. Sendo assim, o objetivo deste trabalho é utilizar métodos de aprendizado de máquina para descobrir, classificar e armazenar textos de notícias falsas para posterior aplicação a ETL de um Data Warehouse e um ambiente de consulta que contribuirá com pesquisas futuras. Para isso, foi criado um dataset, e os métodos Regressão Logística, Naive Bayes e SVM foram avaliados. Finalizando, o trabalho conta com a seleção do melhor método, que foi inserido em um sistema de avaliação online..*

1. Introdução

Desde o início da Web, o volume de dados que estão nos repositórios na rede mundial tem crescido de forma exponencial. Atualmente, são cerca de 200 milhões de sites ativos na Internet, dos quais apenas a rede social *Twitter* gera, em média, 500 milhões de postagens por dia. Tal explosão de dados levou a um estudo do IDC (*Institute Data Corporation*) que estima que até 2020 serão gerados 44 *zettabytes* de dados em todo mundo (IDC, 2012).

Nos diferentes nichos de redes sociais que surgiram, observou-se maneiras diferentes de redigir críticas, propiciadas pelas características das aplicações. Sites específicos, como especializados em críticas de filmes, permitem que usuários escrevam textos relativamente longos. Os microblogs, por outro lado, impõem limites na quantidade de caracteres das mensagens e não são ambientes exclusivamente destinados

para publicação de críticas. No processo de descoberta e pesquisa que prosseguiu nas redes sociais, surgiu a necessidade de expressar opiniões de forma mais direta. (VON LOCHTER, 2015).

Segundo Nogueira (2018), os sites de notícias são o terceiro maior veículo de informação mais acessado da Internet, perdendo apenas para aplicativos de mensagens e redes sociais. Esta informação reflete a importância do uso de sites de notícias e seu impacto no cotidiano das pessoas. Juntoa importância de textos de notícias e seu compartilhamento das mesmas em redes sociais, vem a ascensão e disseminação das notícias falsas. Desde meados de 2017, a quantidade de eventos e debates acerca deste fenômeno que vem sendo chamado de *fake news* cresceu de forma exponencial. *Fake news* pode ser definida como artigos de notícias que são intencional e verificadamente falsos e podem enganar os leitores. Nessa definição de *fake news* inclui artigos de notícias fabricados intencionalmente, como um artigo amplamente compartilhado do agora extinto site *denverguardian.com* com a manchete "*FBI agent suspected in Hillary e-mail leaks found dead in apparent murder-suicide*"(Agente do FBI suspeito de vazamento de e-mail de Hillary encontrado morto em aparente assassinato-suicídio) (DELMAZO, 2017).

Diante da facilidade com que hoje em dia qualquer pessoa pode ter acesso à informação, e com a facilidade do seu uso, vivenciamos uma era de grandes avanços e soluções, seguido porém, por problemas ainda maiores, como é o caso das notícias falsas. Segundo Monteiro et al. (2018), devido à sua natureza atraente, as notícias falsas se espalham rapidamente, influenciando o comportamento das pessoas em diversos assuntos, desde questões saudáveis (por exemplo, revelando medicamentos milagrosos) até política e economia (como no recente escândalo Cambridge Analytica / Facebook e na situação Brexit). Almejando contribuir com tais pesquisas, este trabalho tem como objetivo acoplar à etapa de ETL (Extract, Transform, Load) de um Data Warehouse de Notícias o enriquecimento semânticoatravés de classificação do tipo de notícias: real ou falsa, bem como uma ferramenta de consulta online.

2. Trabalhos Correlatos

No que se refere a notícias falsas e a aplicação de *Machine Learning* Gruppi et al. (2018) construíram um *dataset* com notícias, em português e inglês, tendo por objetivo construir um classificador para predizer se a fonte da notícia é ou não confiável. Utilizando um algoritmo de SVM com um *kernel* linear, foi possível estabelecer as características mais importantes, bem como sua classificação. Como resultado, o algoritmo de classificação obteve acurácia de 85% para os *datasets* brasileiros e 72% para *datasets* Americanos. Em uma contribuição para a área de classificação de notícias, Monteiro, et al. (2018) utilizam o *dataset* Fake.br com o objetivo de avaliar os principais métodos de pré-processamento de textos para avaliar o desempenho do método SVM. Os melhores resultados foram obtidos com a combinação de *bag-of-words* com sentimentos, bem como o uso de todos os atributos, ambos com acurácia de 90%.

Marumo (2018) coletou notícias de sites com notícias verídicas e sites com notícias falsas e/ou de cunho satírico, com o objetivo de encontrar o melhor método para detecção de f*ake news*. Como parte do pré-processamento dos dados, utilizou-se o *framework Gensim* para remoção de caracteres não alfabéticos, a substituição de espaçamentos e quebra de linhas para espaços únicos, remoção de palavras com menos de 3 caracteres e a conversão de letras maiúsculas para minúsculas. Também foi utilizado o *framework keras* para *tokenização* dos dados. Com a aplicação dos

algoritmos de classificação LSTM e SVM, conseguiu-se uma acurácia acima de 90%.

No que se refere ao enriquecimento semântico em ambientes de Data Warehouse através do emprego de técnicas de *Machine Learning*, é o caso Mansman (2014), que obteve um modelo multidimensional da rede social Twitter e desenvolveu um ambiente de Data Warehouse que permitiu a criação de um cubo de dados, bem como a análise de sentimentos. Nogueira (2018), em uma abordagem similar, desenvolveu um ambiente de Data Warehouse que coletas notícias em inglês em tempo real, no qual após avaliação regressão logística, Naïve Bayes, SVM e Perceptron tiveram resultados próximos, dos quais o este último foi utilizado para realizar o enriquecimento semântico na etapa de ETL.

Overfitting constitui-se um grande problema em se tratando de base de dados textuais. Sendo assim, Feng, et. al. (2017), utilizaram o algoritmo *AdaBoost*, conhecido por obter grande sucesso para redução de overfitting em detecção de faces, reconhecimento de caracteres (OCR) e classificação de veículos. Em seus experimentos, foram utilizados datasets de 20 grupos de notícias, dataset Reuters, que consiste em 22 arquivos com um total de 21,758 documentos, e um dataset da BioMed, o qual é dividido em 10 tópicos, cada um contendo entre 1966 e 5022 artigos. Os resultados foram uma média de 86% de acurácia no algoritmo *AdaBoost* (*Bonzaiboost*).

3. Desenvolvimento

Após pesquisas por base de dados com fake news, verificamos que existem poucos recursos disponíveis no idioma Português do Brasil, no qual o dataset mais utilizado é o Fake.br (MONTEIRO, et al., 2018). A proposta apresentada, tem como objetivo proporcionar um ambiente com dados consistentes e limpos na forma de um corpus multidimensional para consumo por aplicações externas e usuários. O corpus multidimensional é um conjunto de textos armazenados de acordo com um modelo multidimensional, que permite explorar a multidimensionalidade em diferentes níveis de abstração: tempo, categoria das notícias, tipo (verdadeira ou fake news).

A metodologia deste trabalho é baseada na arquitetura proposta por Nogueira (2018), na qual o classificador gerado será acoplado a etapa de ETL de um Data Warehouse gerando o enriquecimento semântico em uma nova dimensão.

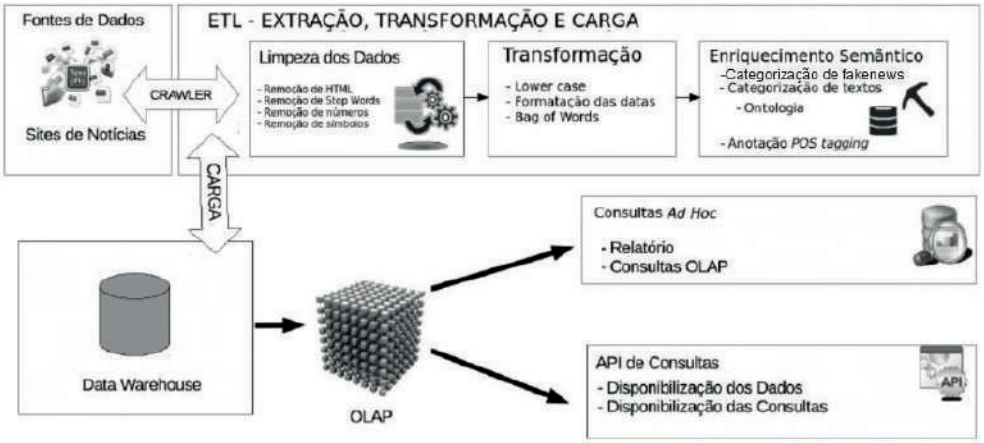


Figura 1. Arquitetura proposta, adaptado de Nogueira (2018)

Para realizar os experimentos foi desenvolvido um web crawler, utilizando a linguagem python, juntamente com a biblioteca *beautiful soup* para a coleta inicial dos dados. Foi construído um dataset composto por 1744 títulos e corpo de notícias falsas coletadas dos sites *boatos.org* e *g1.globo.com/fato-ou-fake,* e 3185 títulos e corpo de notícias verdadeiras coletadas do site *brasil.elpais.com*. Inicialmente será efetuado testes utilizando apenas os títulos das notícias, posteriormente o corpo junto ao título e fazer um comparativo entre ambos. Para isso, serão utilizados os algoritmos de aprendizado de máquina (*Machine Learning*), Regressão Logística, *AdaBoost, Naive Bayes* e SVM.

A Regressão Logística (*Logistic Regression*) é uma técnica estatística que tem como objetivo produzir, a partir de um conjunto de observações, um modelo que permita a predição de valores tomados por uma variável categórica, frequentemente binária, a partir de uma série de variáveis explicativas contínuas e/ou binárias. É útil para modelar a probabilidade de um evento ocorrer como função de outros factores. A regressão logística analisa dados distribuídos binomialmente da forma

$$Yi \sim B(pi, ni), \text{ for } i = 1, \ldots, m, \quad (1)$$

onde os números de ensaios de Bernoulli ni são conhecidos e as probabilidades de êxito pi são desconhecidas. Um exemplo desta distribuição é a percentagem de sementes (pi) que germinam depois de ni serem plantadas.

O *AdaBoost* é um algoritmo meta-heurístico, e pode ser utilizado para aumentar a performance de outros algoritmos de aprendizagem. O *AdaBoost* chama um classificador fraco repetidamente em iterações

$$t = 1, \ldots, T \quad (2)$$

Para cada chamada a distribuição de pesos Dt é atualizada para indicar a importância do exemplo no conjunto de dados usado para classificação. A cada iteração os pesos de cada exemplo classificado incorretamente é aumentado (ou alternativamente, os pesos classificados corretamente são decrescentes), para que então o novo classificador trabalhe em mais exemplos.

No aprendizado de máquina, classificadores *Naive Bayes* são uma família de simples “classificadores probabilísticos” baseados na aplicação do teorema de Bayes com pressupostos de independência fortes (naive) entre as características. Os classificadores *Naive Bayes* são altamente escalonáveis, exigindo um número de parâmetros lineares no número de variáveis (recursos/preditores) em um problema de aprendizado. O treinamento de máxima verossimilhança pode ser feito através da avaliação de uma expressão de forma fechada, que leva um tempo linear, ao invés de uma aproximação iterativa dispendiosa como usada para muitos outros tipos de classificadores. Abstratamente, é um modelo de probabilidade condicional: dada uma instância de problema a ser classificada, representada por um vetor

$$x = (x1, \ldots, xn) \quad (3)$$

representando alguns n recursos (variáveis independentes), atribui a esta instância probabilidades

$$p(Ck \mid x1, \ldots, xn) \quad (4)$$

para cada um dos *K* possíveis resultados ou classes *Ck* .

O *Support Vector Machine* (SVM), também conhecido como Máquina de Suporte Vetorial, foi elaborado com o estudo proposto por Boser, Guyon e Vapnik (1992). É um algoritmo de aprendizado supervisionado, cujo objetivo é classificar determinado conjunto de pontos de dados que são mapeados para um espaço de características multidimensional usando uma função kernel, abordagem utilizada para classificar problemas. Nela, o limite de decisão no espaço de entrada é representado por um hiperplano em dimensão superior do espaço. No caso do kernel linear, recebemos um conjunto de dados de treinamento de *n* pontos da forma

$$(x1,y1),...,(xn,yn), \quad (5)$$

onde os *yi* são 1 ou -1, cada um indicando a classe à qual o ponto *xi* pertence. Cada *xi* é um *p*- vetor real tridimensional. Queremos encontrar o "hiperplano de margem máxima" que divide o grupo de pontos *xi* para qual *yi = 1* do grupo de pontos para os quais *yi = -1*, que é definido de modo que a distância entre o hiperplano e o ponto mais próximo *yi* de qualquer um dos grupos é maximizado.

A partir da criação de um sistema de coleta, com um algoritmo acoplado à etapa de ETL, este irá automaticamente classificar os dados coletados, aumentando assim a acurácia do classificador, e gerando uma base maior de dados para futuros trabalhos de combate a *fake news*. Os quais, inicialmente, devem ser analisados por um humano antes de serem incorporados novamente no sistema. Também foi construído uma interface *Web*, onde o usuário será capaz de submeter um link e verificar se este é ou não uma notícia verdadeira, servindo este como protótipo antes de ser submetido a etapa de ETL (sendo esta, o propósito geral deste trabalho).

4. Resultados Parciais

Os dados obtidos receberam tratamento de valores nulos, ruídos (caracteres especiais, tais como vírgulas, pontos, parênteses, etc) e transformação para letras minúsculas. Cada dataset recebeu uma nova coluna, chamada label, onde foi atribuído o valor booleano 0 para notícias verdadeiras, e 1 para as notícias falsas. Com isso, os dados foram combinados em um único dataset.

Inicialmente, o *dataset* utilizado continha apenas os títulos das notícias, sendo então dividido entre treino e teste, na proporção de 75% e 25% respectivamente. A primeira parcela serve para treinar o algoritmo, enquanto a segunda, para verificar a acurácia do mesmo. Na sequência, receberam tratamento de tokenização, utilizando o pacote NLTK, com o *bag of words* em português do Brasil. Testes efetuados utilizando os algoritmos Regressão Logística (Logistic Regression), *AdaBoost*, Naive Bayes e SVM (kernel linear), obtiveram a acurácia de 88,85%, 81,37%, 86,22% e 87,45% respectivamente, no modelo de testes. Como técnica de avaliação dos modelos empregados, foi utilizado a validação cruzada com o método k-fold = 10.

Novamente o dataset foi dividido entre treino e teste, juntando agora os títulos ao corpo das notícias. Receberam os mesmos tratamento acima citados, obtendo a acurácia de 90,88%, 84,23%, 91,19% e 91,16% nos algoritmos Regressão Logística (*Logistic Regression*), *AdaBoost*, *Naive Bayes* e SVM respectivamente. A aplicação do método de validação cruzada, revelou um *overfitting* em alguns casos. Por fim, o *dataset* foi dividido para utilização apenas dos corpos das notícias. Foram empregados os mesmos métodos utilizados anteriormente em relação ao tratamento e limpeza dos dados. A

aplicação dos algoritmos resultou em 90,88%, 94,23%, 91,19% e 91,16% de acurácia nos algoritmos Regressão Logística (*Logistic Regression*), *AdaBoost*, *Naive Bayes* e SVM respectivamente (Tabela 1).

Tabela 1. Comparativo entre os datasets em relação à acurácia e validação cruzada.

	Regressão Logística	AdaBoost	Naive Bayes	SVM (kernel Linear)
Título	88,85%	81,37%	86,22%	87,45%
K-fold	0,88	0,75	0,86	0,55
Corpo	97,40%	95,12%	97,80%	98,62%
K-fold	0,97	0,95	0,97	0,64
Título + Corpo	90,88%	84,23%	91,19%	91,16%
K-fold	0,90	0,84	0,91	0,54

A partir da análise de resultados, o método de Naive Bayes foi selecionado o melhor método, pelo fato de obter uma alta acurácia, complementado de ser um método de aprendizado incremental (online). Posterior ao acoplamento foi desenvolvido a interface de classificação de fake news, mostrada pela Fig 2. e está disponível no servidor <https://detectorfakenews.herokuapp.com>. A ferramenta espera como parametro o link de um site de notícia, e retorna se ele é ou não uma notícia falsa (*fake news*)

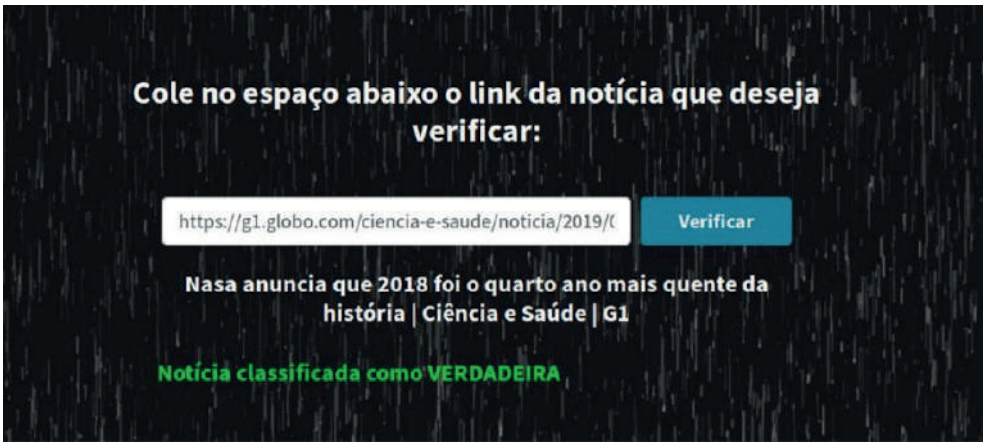


Figura 2. Interface Web da Aplicação desenvolvida. Disponível em: <https://detectorfakenews.herokuapp.com/>. Acesso em 18 ago. 2019.

5. Considerações Finais e Trabalhos Futuros

O *overfitting* constitui-se um problema recorrente em bases textuais. Alguns algoritmos chegaram a resultados bastante relevantes, mas ao aplicarmos a validação cruzada com k=10, notou-se um grande *overfitting* em alguns casos. Sendo assim, observou-se que o algoritmo *Naive Bayes* obteve além da alta acurácia, tolerância ao *overfitting*, sendo este escolhido para implementação inicial na ferramenta online.

Para futuros trabalhos, tem-se como objetivo avaliar outras características técnicas de pré-processamento, aumentar a base de treino, aplicar os novos resultados a interface *web*, e posteriormente, o acoplamento a ETL do *Data Warehouse*.

Referências

DELMAZO, Caroline; VALENTE, Jonas CL. Fake news nas redes sociais online: propagação e reações à desinformação em busca de cliques. Media & Jornalismo, v. 18, n. 32, p. 155-169, 2018.8

FENG, Xiaoyue; LIANG, Yanchun; SHI, Xiaohu; XU, Dong; WANG, Xu; GUAN, Renchu. “Overfitting Reduction of Text Classification Based on AdaBELM”, 2017

FREUND, Yoav; SCHAPIRE, Robert E. “A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting”, 1995

GRUPPI, Maurício; HORNE, Benjamin D.; ADALI, Sibel. “An Exploration of Unreliable News Classification in Brazil and The U.S.” Rensselaer Polytechnic Institute, Troy, New York, USA.2018.

IDC. Gantz, J., & Reinsel, D. (2012). The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. IDC iView: IDC Analyze the future, 2007(2012), 1-16.

GARSON, David G (?).Logistic Regression: Statnotes, from North Carolina State University, Public Administration Program. Disponível em: <https://faculty.chass.ncsu.edu/garson/PA765/statnote.htm> Acesso em 31 de maio de 2019

MANSMANN, Svetlana; REHMAN, Nafees Ur; WEILER, Andreas; SCHOLL, Marc H. “Discovering OLAP dimensions in semi-structured data.” Information Systems, v. 44, p.120-133, 2014.

MARON, M. E. (1961). "Automatic Indexing: An Experimental Inquiry" (PDF). Journal of the ACM. 8 (3): 404–417.

MARUMO, Fabiano Shiiti. “Deep Learning para classificação de Fake News por sumarização de texto.” - Londrina, 2018.

MONTEIRO, Rafael A.; SANTOS, Roney L. S.; PARDO, Thiago A. S.; ALMEIDA, Tiago A. de; RUIZ, Evandro E. S.; VALE, Oto A.. “Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results.” In: International Conference on Computational Processing of the Portuguese Language. Springer, Cham, 2018. p. 324-334.

NARASIMHA Murty, M.; SUSHEELA Devi, V. (2011). Pattern Recognition: An Algorithmic Approach.

NOGUEIRA, Rodrigo Ramos. O Poder do Data Warehouse em Aplicações ed Machine Learning: Newsminer: Um Data Warehouse Baseado em Textos de Notícias. São Paulo: Nea, 2018.

RUSSELL, Stuart; NORVIG, Peter (2003) [1995]. Artificial Intelligence: A Modern Approach (2nd ed.). Prentice Hall.

VON LOCHTER, Johannes et al. Máquinas de classificação para detectar polaridade de mensagens de texto em redes sociais. 2015.

Desenvolvimento de um sistema de análise de sentimento utilizando técnicas de Data Warehousing

Jonathan Suter, Rodrigo Ramos Nogueira, Leonardo Croda, Daniel Anderle

Instituto Federal Catarinense – Campus Camboriú (IFC) – Camboriú, SC – Brasil

jonathan.vinicius.suter@gmail.com; rodrigo.nogueira@ifc.edu.br;
lccroda@gmail.com, daniel.anderle@ifc.edu.br

Abstract. *Twitter is one of the most used social networks in the world, in which a user can be translated as a whole and published worldwide. Consuming data from this social network, the texts were classified as "feelings" (good, ruins and neutral). From the classified data, a data warehouse system was developed, which coupled a machine learning algorithm. And so were the most texts, 107393, totaling the 108693 tweets. From the data collected and classified, an analysis of the data of the presidential election of 2018 was made.*

Resumo.*O twitter é uma das redes sociais mais utilizadas do mundo, no qual um usuário pode ser traduzido como um todo e publicado em todo mundo. Consumindo dados dessa rede social, os textos foram classificados como "sentimentos" (bons, ruínas e neutros). A partir dos dados classificados, foi desenvolvido um sistema de data warehouse, que acopla um algoritmo de aprendizado de máquina. E assim foram os os mais textos, 107393, totalizando os 108693 tweets. A partir dos dados coletados e classificados, foi feita uma análise dos dados da eleição presidencial de 2018.*

1. Introdução

Dentre diversas aplicações em um *corpus* linguístico baseado em textos do *Twitter,* se destacam as pesquisas que explorar a análise de sentimento. O processo de análise de sentimentos consiste na abordagem computacional que, com a utilização de técnicas de processamento de linguagem natural e aprendizagem de máquina, tem o objetivo de julgar textos a fim de determinar sentimentos e opiniões presentes em frases. Análise de sentimentos também é comumente conhecida por vários outros termos, tais como: extração de opinião, mineração sentimento, análise de subjetividade, análise afetiva, análise de emoções e mineração de opinião (JUNQUERA,2017).

As redes sociais tem grande importância para a sociedade está relacionada ao fato de que as mesmas possuem grande potencial de compartilhamento de informação. Sendo assim, os dados extraídos de uma rede social, podem ser utilizados para o auxílio na tomada de

decisão de determinado assunto de cunho estratégico, para uma corporação ou até mesmo um indivíduo (TOMAÉL ,2005).

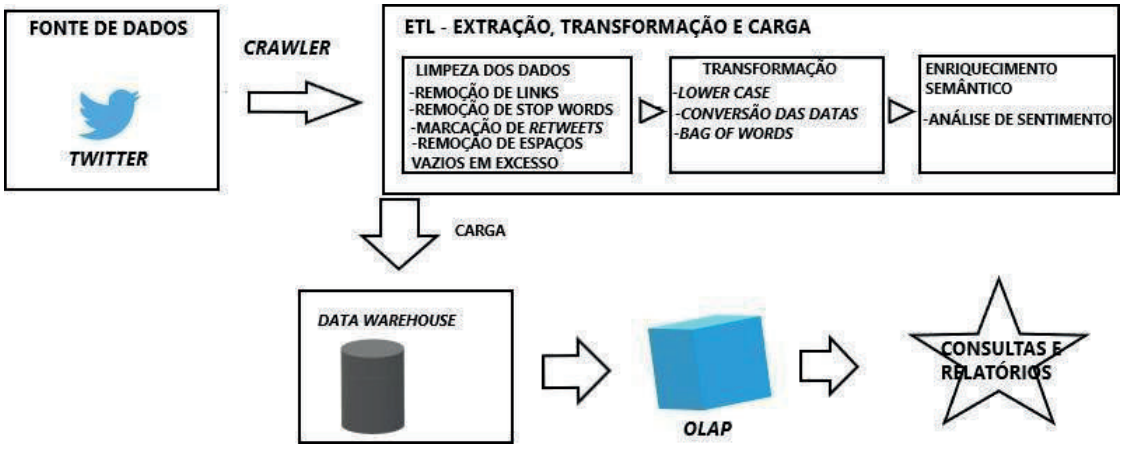
No entanto, por mais interessante que seja a aplicação de aprendizado de máquina para extração de sentimento, o grande desafio no emprego de técnicas de aprendizado de máquina é que 80% de todo o esforço computacional é gasto na etapa de pré-processamento de dados (LOSARWAR, 2012). O desenvolvimento de uma ferramenta que faça a coleta dos dados, realize a limpeza, normalize os mesmos e guarde-os em uma estrutura definida, além de diminuir o esforço nesta etapa, ainda facilita a utilização destes dados por terceiros, permitindo ao usuário que foque-se em sua atividade principal de análise destes dados.

A partir dessa problemática, essa pesquisa tem como objetivo o desenvolvimento de um Data Warehouse alimentado com dados em tempo real da rede social *Twitter*, sob o qual foram coletados e analisados os textos sobre a eleição de 2018.

2. Metodologia

Uma vez que o produto final da pesquisa é um conjunto de arquitetura de software, complementada de um conjunto de dados, esta pesquisa se enquadra como pesquisa tecnológica (JUNIOR et al. 2014). O desenvolvimento teve como base na arquitetura de um *Data Warehouse* de KIMBAL(2011). A Figura 1 mostra a arquitetura proposta por esta aplicação de *Data Warehouse*. Para que ocorra o armazenamento dos *Tweets* para posterior uso nas consultas, é efetuada a coleta dos textos assim como o pré-processamento, compondo a etapa de ETL. Finalmente, após os dados pré-processados e limpos podem ser realizadas consultas OLAP para explorar o cubo de dados.

Figura 1. Fluxo de funcionamento da aplicação

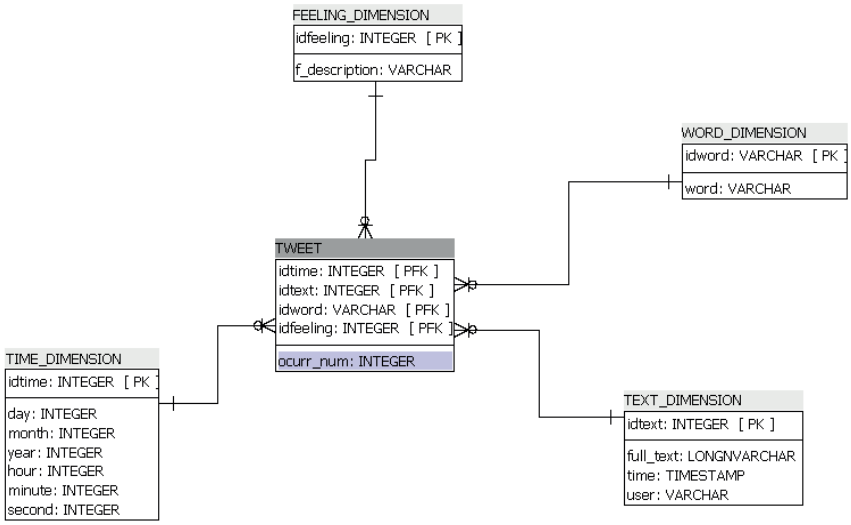


Fonte: Adaptado de Nogueira(2017)

Com os textos já limpos, seleciona-se a data do registro e é efetuada sua formatação para que possa ser inserida na base. A partir disso, os dados do *Tweet* estão preparados para que o mesmo possa "quebrado" e se efetue a *Bag of Words*. Com os dados do *Tweet*, as palavras são quebradas pelo script e inseridas na base de dados multidimensional. Caso a palavra já exista na base, é apenas atualizada sua frequência. E assim, tem se um documento com os termos e sua frequência em cada *Tweet* e com uma consulta, sua frequência na base como um todo.

O modelo multidimensional, mostrado pela Figura 2, foi implementado com o SGDB PostgreSQL, versão 10.2.1 em 64 bits, com o pgAdmin4. Este Data Warehouse é um ROLAP (Relational On-line Analytical Processing), pois seus dados derivam de uma base dados relacional, são uma fração selecionada de dados de uma base relacional, reorganizada. A tabela "TWEET" é a tabela fato as outras são as dimensões, a tabela que "une as demais". A tabela "TEXT_DIMENSION" é a tabela que armazena os textos dos Tweets, o usuário e o momento de criação do Tweet. Na tabela "TIME_DIMENSION" ficam inseridos todas as combinações dos segundos, minutos, horas dias, meses e o ano que existem entre os meses de Julho e Outubro. "WORD_DIMENSION" é a tabela em que ficam registrados os termos extraídos dos textos dos Tweets. A tabela "FEELING_DIMENSION" é responsável por armazenar os sentimentos, no caso, "Positivo", "Neutro" e "Negativo".

FIGURA 2. Modelo multidimensional desenvolvido



Fonte: Os autores.

Foram coletados 108893 *Tweets* entre os meses de julho e outubro, referentes à *hashtag* "eleicoes2018". Após as etapas de coleta, preparação dos textos e enriquecimento semântico e, ao efetuar o treinamento do algoritmo de classificação, usando o conjunto de dados para treinamento com 1300 *tweets* classificados

3. Resultados e Discussões

Esta seção tem como objetivo ilustrar alguns dos resultados obtidos que podem ser explorados a partir do modelo multidimensional desenvolvido. Para a execução das consultas, foi gerada uma base com dados coletados entre setembro e outubro de 2018 utilizando como termo de busca "eleicoes2018". Vale ressaltar que a API não confere acesso total à base de dados da rede social e há limite de coleta por dia. A começar pela palavra de maior menção.

Tabela 2. Ranking de palavras com maior número de ocorrências

Palavra	Quantidade
eleições2018	51458
bolsonaro	24424
candidato	10559
haddad	9726
diz	8184
presidente	7188
contra	7160
eleições	7125
sobre	6443

Fonte: Os autores.

Pode-se observar que naturalmente, o termo usado para a pesquisa dos *Tweets* é o que tem mais ocorrências, este pode ser desconsiderado no momento. Entretanto, a segunda palavra

mais citada entre os textos é “bolsonaro”. O segundo termo mais citado é “candidato” e o terceiro é “haddad”, indicando primariamente que estes foram os candidatos mais citados.

Ao analisar as menções diretas por candidato, os valores foram Bolsonaro:24424, Haddad:9726, Ciro:4510, Alckmin:1897, Daciolo:1819, Marina: 1817, Boulos:1098, Meirelles: 588, Amoêdo: 192, Álvaro:139, Goulart: 78,Vera:61, Eymael:13.

No primeiro momento, é possível observar que nenhum candidato obteve mais citações boas que ruins, refletindo que o sentimento geral entre os *tweets* foi ruim, e que nenhum candidato conseguiu obter uma grande aprovação dos eleitores.Com base nos textos. O candidato que obteve a maior quantidade de citações com sentimento “bom” foi o Bolsonaro. Entretanto, também foi o candidato que obteve a maior quantidade de citações classificadas como “ruim”.

Ao comparar os resultados com os da eleição (Disponível em <https://especiais.gazetadopovo.com.br/eleicoes/2018/resultados/votacao-candidatos-presidente-brasil/>), Obtém-se certa equivalência entre os resultados extraídos do *Data Warehouse* e as intenções de voto, apesar de muitas divergências, há de se considerar ainda que a base possui muitas citações qualificadas como neutras, podendo ocorrer maior distribuição para as citações com sentimento “ruim” e “bom”.

3. Considerações Finais

O tratamento e análise de textos escritos por pessoas, que possuem pouca ou nenhuma revisão, ainda mais em um espaço de informalidade como o Twitter, podem trazer desafios, tanto com os dados em si quanto com o sentido que eles possuem. Assim, a inserção de uma etapa para classificação dos textos como parte da ETL se tornou essencial para automatizar essa tarefa, que pode ser bastante morosa para um humano. Desta forma, o pré-processamento dos textos para que os mesmos possam entrar na base de dados já limpos e qualificados permite ao usuário se preocupar apenas com o processo analítico dos dados, e desta forma, extrair informações e relatórios, como proposto. Apesar das limitações que a API do Twitter impõe, ainda é possível criar aplicações interessantes, usando os métodos corretos para a estrutura e análise dos dados. As consultas efetuadas para explorar o modelo multidimensional do Data Warehouse são só alguns exemplos do que pode ser feito.

As consultas efetuadas e os dados extraídos, foram capazes de demonstrar bem o sentimento dos eleitores a respeito das eleições como um todo e dos candidatos. Muita indiferença dos eleitores em relação às eleições; grande parte das pessoas que possuíam algum sentimento em relação aos candidatos, levaram para o Twitter o sentimento geral sobre os políticos: desaprovação, seja por ações ou ideologias de cada. O fato é que, a amostra deste estudo e sua análise é coerente até certo ponto com os fatos verificados no mundo real, gerando a necessidade de melhorias na aplicação com um todo.

REFERÊNCIAS

JUNIOR, Vanderlei FREITAS et al. **A pesquisa científica e tecnológica**. Espacios, v. 35, n. 9, 2014.

JUNQUEIRA, Kássio TC; DA ROCHA FERNANDES, Anita Maria. **Análise de Sentimento em Redes Sociais no Idioma Português com Base em Mensagens do Twitter**. Anais do Computer on the Beach, p. 681-690, 2018.

KIMBALL, Ralph; CASERTA, Joe. **The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data**. John Wiley & Sons, 2011.

LOSARWAR, V.; JOSHI, D. M. **Data preprocessing in web usage mining**. In: Proceedings of International Conference on Artificial Intelligence and Embedded Systems. New Asia, Singapura: [s.n.], 2012. p. 15–16.

NOGUEIRA, Rodrigo Ramos, et al**. Newsminer: um sistema de data warehouse baseado em texto de notícias.** 2017.

TOMAÉL, Maria et. al. **Das redes sociais à inovação.**Ciência da Informação. Brasília: 2005. Volume 34, numero 2. p. 93-104.

Descoberta de Perfis de *Youtubers* via Aprendizagem de Máquina

Anderson de Alencar Bezerra Souza, Ticiana L. Coelho da Silva, Matheus Oliveira Costa, Pedro Henrique Pereira, Georgia Cruz, Andrea Pinheiro

[1]Instituto Universidade Virtual. Fortaleza - Ceará, Brasil

{anderson.sjav, matheusoliveiracos}@gmail.com,

{pedrohenripereira}@alu.ufc.br,

{ticianalc, georgia, andrea}@virtual.ufc.br

Resumo. *O uso progressivo de redes sociais, como o YouTube, nos últimos anos tem produzido um grande volume de informações geradas pelos seus usuários, que com frequência compartilham seus sentimentos, opiniões sobre vídeos, bem como novos conteúdos em formato de mídias (vídeos). Compreender o perfil desses usuários produtores de conteúdo, mais conhecidos como youtubers, pode ser um diferencial para sistemas de recomendação que podem sugerir canais na plataforma YouTube para seus usuários com mais precisão, e ainda, auxiliar no entendimento de como a comunicação falada de crianças e adultos está mudando com o passar dos anos. No entanto, analisar esse grande volume de dados de forma não automatizada consiste em um problema não trivial. Este trabalho visa descobrir e identificar os perfis de Youtubers utilizando técnicas de Aprendizagem de Máquina. Foi realizado um estudo experimental com 286 vídeos obtidos a partir da plataforma YouTube para avaliar nossa proposta.*

1. Introdução

As novas aplicações da Web 2.0 permitem aos indivíduos criarem conteúdo e colocarem na rede para outros verem e interagirem de forma praticamente livre e fácil. Um dos maiores representantes dessa nova realidade é o *site* de compartilhamento de vídeos YouTube.

Criado em 2005, o YouTube permite que qualquer pessoa possa enviar, assistir e interagir com outros usuários na sua rede, de forma gratuita e simples. Esse diferencial possibilitou milhares de pessoas a criarem e disponibilizarem conteúdo original *online*. O *site* também conta com um complexo sistema de recomendação, que se baseia no perfil de uso dos usuários para indicar vídeos relacionados.

No entanto, não existe algo de mesma magnitude no caso de recomendação de *youtubers* e também não é de conhecimento dos autores nenhuma ferramenta que execute a ação de relacionar vídeos aos seus criadores de forma a classificar e agrupar tais vídeos de acordo com o perfil do *youtuber*. Isso pode ser relevante para sistemas de recomendação de canais ou de *youtubers* aos usuários da plataforma. E ainda, auxiliar no entendimento de como a comunicação falada de crianças e adultos está mudando com o passar dos anos. Segundo a Oxford, *youtuber* é um usuário frequente do *site* de compartilhamento de vídeos YouTube, especialmente alguém que produz e aparece em vídeos na plataforma.

Este trabalho visa propor uma solução por meio da análise dos textos utilizando técnicas de aprendizagem de máquina nas legendas dos vídeos do YouTube, de tal sorte que canais ou *youtubers* de alta similaridade (ou mesmo perfil) estejam em agrupados em um mesmo perfil ou grupo. Bem como, canais ou *youtubers* dissimilares estejam em grupos diferentes.

As técnicas de mineração de dados [Han et al. 2011] serão utilizadas na identificação dos grupos mencionados anteriormente. Elas são usadas em muitas aplicações em diferentes contextos, inclusive análise de conteúdo online. Trabalhos anteriores demonstram a utilização dos métodos para descobrir perfis extremistas em blogs [Chau and Xu 2007] e vídeos [Sureka et al. 2010], para o reconhecimento de padrões em vídeos baseados nos textos derivados deles [Chang et al. 2005] e também para agregar perfis de uso para uma experiência personalizada [Mobasher et al. 2002].

Os resultados desta pesquisa, além de auxiliarem na descoberta dos perfis de *youtubers*, servirão como base para construção de uma base de treinamento para um classificador de vídeos, capaz de categorizar um dado vídeo, se ele se refere a um conteúdo para crianças, para amantes de jogos, maquiagem, entre outros perfis.

O presente trabalho está dividido da seguinte forma: a Seção 2 apresenta os principais conceitos para entendimento deste trabalho. A Seção 3 discute brevemente a proposta deste trabalho. A Seção 4 apresenta um estudo experimental realizado com 286 vídeos extraídos da plataforma do YouTube. A Seção 5 discute as conclusões e trabalhos futuros.

2. Fundamentação Teórica

Nesta seção, são discutidos os principais conceitos envolvidos neste trabalho.

2.1. Clusterização

Clusterização refere-se a um conjunto de técnicas de aprendizagem de máquina que pode ser definida como o processo de dividir grandes conjuntos de dados em subconjuntos que apresentam similaridades entre si e diferenciação entre os outros subconjuntos, chamados *clusters*. A divisão é feita baseada no algoritmo de análise escolhido. Nesse contexto, diferentes métodos de análise podem gerar diferentes agrupamentos no mesmo conjunto de dados. Desse modo, pode-se descobrir outros grupos de dados que apresentam similaridades que não foram observados anteriormente [Han et al. 2011].

O algoritmo K-means é utilizado neste trabalho por sua facilidade de uso e eficiência computacional, ele se comporta da seguinte forma. Inicialmente, é necessário fornecer ao algoritmo um número K clusters que dará origem aos centróides. Os centróides são então dispostos aleatoriamente no conjunto de amostragem e para cada amostra é calculada sua distância euclidiana até o centróide mais próximo, atribuindo a este, aquela amostra. Em seguida, computa-se a média de todas as posições das amostras atribuídas a um dado centróide e aos centróides é atribuída uma nova posição a partir do resultado dessa média, colocando-os ao centro delas.

Repete-se o cálculo, da distância euclidiana ao centróide mais próximo para cada amostra e calcula-se novas posições dos centróides até que o número de iterações tenha se esgotado ou até que o erro computado pela distância dos pontos ao centro do centróide seja satisfatótio para o usuário. Na experimentação, foi utilizada a *scikit-learn*, que disponibiliza o algoritmo K-means.

2.2. Representação Vetorial das palavras

Representação Vetorial das palavras ou *Word Embedding* é definido como um conjunto de técnicas para geração de representações vetoriais de palavras, derivadas de vários métodos de treinamento inspirados em modelos neurais de linguagem [Levy and Goldberg 2014]. As representações vetoriais conseguem codificar as previsibilidades sintáticas e semânticas com alta precisão e preservam as regularidades lineares entre as palavras[Mikolov et al. 2013].

O modelo diferencia-se de métodos como o *One-hot Encoding* que armazena vetores binários de grandes dimensões, que apresentam o mesmo tamanho do vocabulário de palavras. Os *Words Embeddings* são vetores de ponto flutuante, portanto com com menor quantidade de valores, normalmente as dimensões utilizadas são de 256, 512 ou 1024. Para cada palavra é gerado um vetor denso no qual, a sua representação vetorial é similar a representação de palavras análogas, que podem aparecer em contextos similares [Chollet 2017]. Deste modo, esperamos que dentro de um espaço vetorial, palavras semelhantes tenham representação vetorial próximas umas das outras.

2.3. Glove

Global Vectors for Word Representation(GloVe) é um algoritmo de aprendizagem não supervisionado para obter representações vetoriais de palavras. Utilizando o modelo de regressão log-bilinear global para a obter as representações e, por isso, supera outros modelos nas tarefas de analogia, semelhança de palavras e tarefas de reconhecimento de entidades nomeadas. Para o mesmo corpus, tamanho de vocabulário e tempo de treinamento, o GloVe supera em performace o Word2vec, pois alcança melhores resultados mais rapidamente [Pennington et al. 2014].

O algoritmo está disponível em código aberto por pesquisadores da Universidadede Stanford (os desenvolvedores originais) e dessa forma possibilitou ser usado para criar representações de palavras em diversos vocabulários ao redor do mundo. Dentre essas, existe a representação disponibilizada pelo Núcleo Interinstitucional de Linguística Computacional da USP (NILC) para a língua portuguesa[1], onde foram avaliados diferentes modelos de incorporação de palavras treinadas em um grande corpus português, incluindo variantes brasileiras e europeias. Foram geradas representações utilizando quatro técnicas de *Word Embedding*, dentre elas o GloVe [Hartmann et al. 2017]. A matriz obtida através do Glove utilizada neste trabalho tem 50 dimensões para cada palavra.

2.4. Método Elbow

O método Elbow (método do cotovelo) auxilia encontrar uma quantidade K ideal de clusters que devem ser utilizados pelo algoritmo de *clusterização*. Este método calcula a média das distâncias euclidianas das amostras, em relação à variação no número de centróides K, a fim de capturar os erros. Quando o erro se estabiliza ou torna-se satisfatório para o usuário, usa-se esse número K de clusters para executar o algorítmo de clusterização.

3. Proposta

O presente trabalho busca descobrir e identificar perfis de canais ou *youtubers* do *site* de compartilhamento de vídeos *online* YouTube por meio do uso de técnicas de clusterização.

[1] http://nilc.icmc.usp.br/embeddings

O processo é feito primeiramente, por meio de sistemas de coleta de dados e processamento de forma automatizada, para depois a visualização dos resultados e interpretação por meio de profissionais dedicados da área de comunicação.

A partir das legendas de cada vídeo coletado, inicia-se o processo de obter uma representação vetorial para cada texto, a qual será utilizada no processo de clusterização posterior. Para alcançar tal objetivo, primeiramente é obtida as representações vetoriais de cada palavra individualmente, nessa tarefa são utilizadas as representações geradas pelo algoritmo *GloVe*, como explicado na seção anterior.

Com as representações das palavras, é calculada a representação resultante do texto. O cálculo necessário é a média das representações das palavras do texto, ou seja, o somatório de todas as representações vetoriais do texto dividido pela quantidade de palavras. A eficácia da abordagem é demonstrada em trabalhos como o de [Kenter et al. 2016] e [Kenter and De Rijke 2015].

Finalmente, as representações dos textos são agrupadas em *clusters* de acordo com suas similaridades baseadas no cálculo das distâncias entre os objetos. Para realizar a clusterização é utilizado o algoritmo *k-means*, cujo o funcionamento foi detalhado anteriormente. Os *clusters* em formato de nuvens de palavras são entregues a profissionais da área de comunicação para que estes avaliem as mudanças na comunicação falada de crianças e jovens *Youtubers*.

A seguir, é explicado o estudo experimental realizado para avaliar a proposta deste trabalho.

4. Experimentação

Foram selecionados vídeos com diferentes conteúdos e públicos para avaliar a proposta deste trabalho. Os grupos selecionados foram de *Youtubers Gamers* e Infantis que dentro de seus próprios nichos apresentam conteúdos diferenciados.

Inicialmente, foi realizada uma coleta de diversos canais famosos da plataforma YouTube e extraída a transcrição de vídeo de cada um deles, entre eles: Baby Doll Kids, Planeta das Gêmeas Games,Brinquedos KidsToys Brasil, LipaoGamer, Tubalatudo, Canal Bobinho Massinhas, Mileninha Stepanienco, Brincadeira de Criança, PupiGames, Jazzghost, Filha Também Joga, Show do Tiago, Brinquedos e Bonecas, Fran Nina e Bel para meninas, Juliana Baltar,Manoela Antelo, entre outros. O total de textos foi de 286 legendas.

Para tratar os textos, foi utilizada a representação vetorial de palavras, obtida através da representação de 50 dimensões disponibilizada pela base de dados *Glove*, confome já mencionado.

O *corpus* de textos foi percorrido e através do algoritmo *TF-IDF* criou-se um dicionário com as palavras mais relevantes, para eliminar *stop-words* e outros termos menos importantes que produziriam ruído nos resultados. Palavras com frequência maior que 35% dentro do corpus foram eliminadas do dicionário, e consequentemente o vocabulário foi restringido a 12000 termos.

Em cada um dos textos foram selecionadas apenas as palavras que estavam presentes no vocabulário previamente elaborado. Tais palavras foram transformadas em vetores

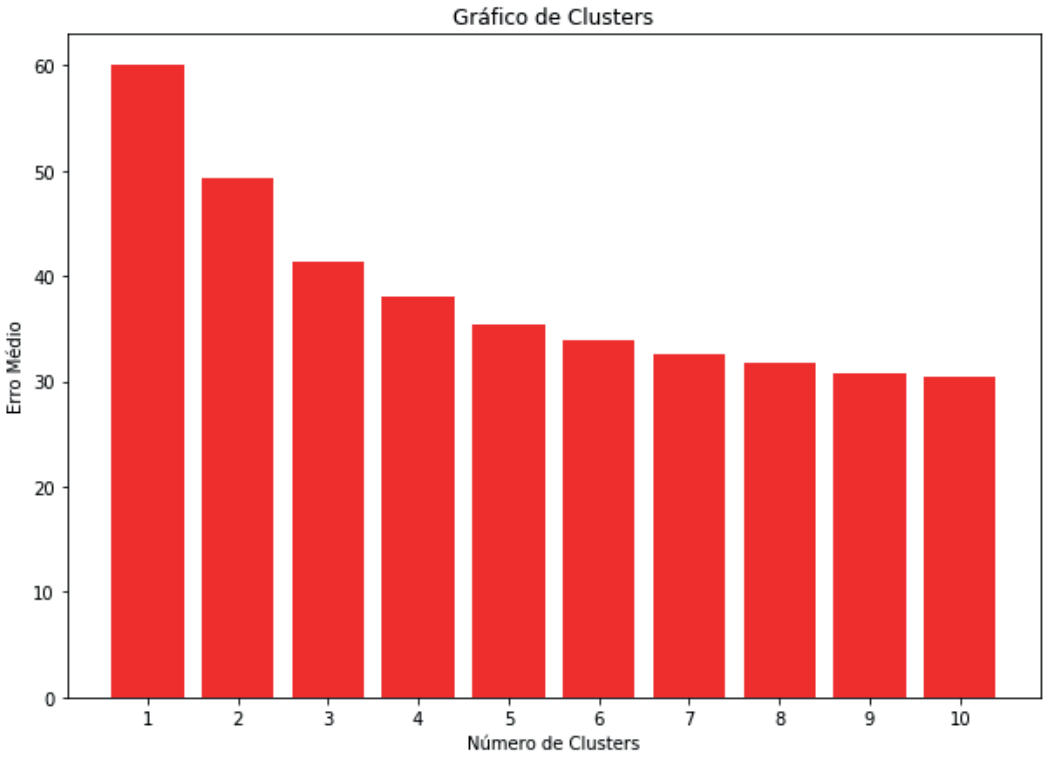


Figura 1. Plotagem do método Elbow

obtidas pelas suas respectivas representações no Glove, formando uma matriz *X* de 50 dimensões do Glove por n palavras presentes no texto para cada transcrição. Os vetores das matrizes *X* são então somados, gerando uma resultante única para o texto. Esse processo se repete para todas as legendas. Esses vetores resultantes são incorporados a uma nova matriz *T* de dimensões: 286 (quantidade total de transcrições ou legendas) por 50 (dimensões do Glove).

É importante ressaltar que algumas palavras dentro da base de dados apresentam dimensões maiores que 50, possuindo, muitas vezes, uma dimensão extra de valor zero que pode originar um erro no momento de fazer soma entre os vetores. Desse modo, quando foi encontrado esse fenômeno, foi removida a dimensão extra na base de dados.

A matriz vetorial *T* foi *clusterizada* utilizando o algorítimo *K-means* escolhendo um número *satisfatório* de *clusters* descoberto por meio do método de Elbow. Para esta experimentação, o erro obtido para K = 5 clusters foi aceitável e suficiente para suspender as iterações seguintes, pois os erros para valores maiores que 5 geravam erros médios de 1 ponto de diferença entre K e K + 1, não justificando a necessidade de mais clusters para o estudo. O gráfico a baixo nos mostra a plotagem do erro para 10 *clusters*.

Logo após, o identificador de cada legenda ou transcrição foi capturado para se identificar a qual *cluster* pertencia e foram geradas nuvens de palavras para as resultantes de mesmo cluster. As Figuras 2 e 3 sintetizam os conteúdos de dois clusters diferentes. A Figura 2 apresenta linguagem mais característica de canais infantis com palavras-chaves como, linda, bonequinha, menina, amiguinho. E a Figura 3 demonstra uma linguagem mais característica do público *Gamer* como, Beleza, Jogo, Carro, Equipe, Mano, Velho.

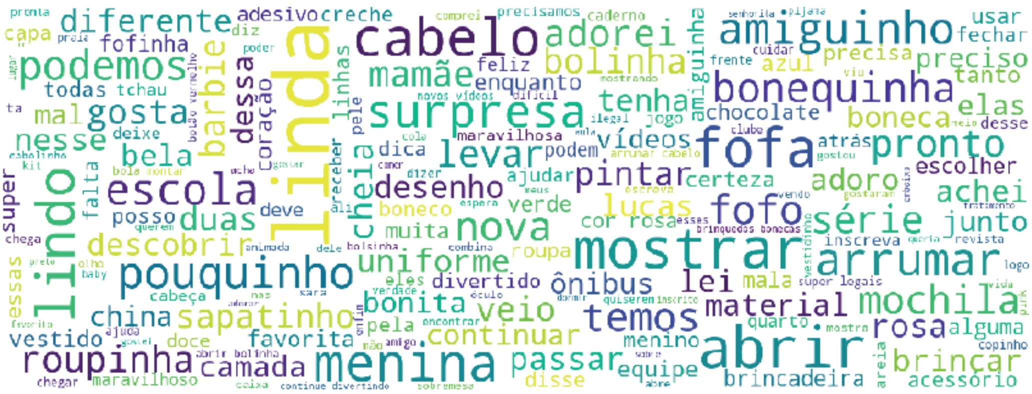


Figura 2. Nuvem de Palavras Cluster I

Figura 3. Nuvem de Palavras Cluster II

Outros clusters mostraram um diferenciação tênue de conteúdo em relação a outros, como por exemplo canais infantis em que os pais gravam os vídeos com seus filhos, palavras-chaves mais amorosas como Mamãe, Filha e Escola apareceram com mais relevância neles como visto na imagem 4.

Como é possível observar cada uma das nuvens de palavras apresenta um grupo de palavras específicas que caracteriza a semelhança dentro do cluster e a diferença entre cada um deles. Com esse método também é possível detectar mudanças no vocabulário dentro de um mesmo nicho, como o destaque para outros grupos infantis em que os pais, podem ou não aparecer modificando o modo de fala. Desse modo é possível observar tipos de linguagens e diferentes formas de se comunicar entre usuários de canais com público e temas discrepantes.

Paralelo as nuvens de palavras foi inspecionado quais vídeos estavam agrupados em cada um dos clusters, de acordo com seus IDs. Percebeu-se que dois clusters tinham concentrações específicas de um nicho de canal.

Em um dos clusters foram agrupados 167 vídeos. O conteúdo é majoritariamente infantil e não trata de temas voltados a jogos. Em um segundo cluster, cujo total de vídeos foi de 78, apresenta narrativas montadas a partir de um jogo, como por exemplo Minecraft. Deve-se levar em conta que o K-means não detecta outliers e a transcrição automática obtida pela API do Youtube não é exatamente igual ao áudio do vídeo.

Figura 4. Nuvem de Palavras Cluster III

5. Conclusão e Trabalhos Futuros

Através dessa pesquisa será possível detectar e estudar padrões e formas de comunicação que vão além das imaginadas socialmente na atualidade. Esta pesquisa pretende ao final do trabalho, disponibilizar uma base de textos das legendas de vídeos do YouTube e ser identificado o algoritmo de clusterização que devolve o conjunto de *clusters* de maior qualidade de acordo com uma métrica a ser escolhida.

Bem como, conforme já dito anteriormente, os resultados desta pesquisa serão utilizados para construir uma base de treinamento para um classificador de vídeos. O papel do classificador será categorizar um dado vídeo, se ele se refere a um conteúdo para crianças, para amantes de jogos, maquiagem, entre outros perfis. Não é do conhecimento dos autores nenhuma abordagem que realize o mesmo processo.

Referências

Chang, S.-F., Manmatha, R., and Chua, T.-S. (2005). Combining text and audio-visual features in video indexing. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–1005. IEEE.

Chau, M. and Xu, J. (2007). Mining communities and their relationships in blogs: A study of online hate groups. *International Journal of Human-Computer Studies*, 65(1):57–70.

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Co., Greenwich, CT, USA, 1st edition.

Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Rodrigues, J., and Aluísio, S. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *CoRR*, abs/1708.06025.

Kenter, T., Borisov, A., and De Rijke, M. (2016). Siamese cbow: Optimizing word embeddings for sentence representations. *arXiv preprint arXiv:1606.04640*.

Kenter, T. and De Rijke, M. (2015). Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1411–1420. ACM.

Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 302–308.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. (2002). Discovery and evaluation of aggregate usage profiles for web personalization. *Data mining and knowledge discovery*, 6(1):61–82.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Sureka, A., Kumaraguru, P., Goyal, A., and Chhabra, S. (2010). Mining youtube to discover extremist videos, users and hidden communities. In *Asia Information Retrieval Symposium*, pages 13–24. Springer.

Uma Análise da Evasão no Ensino Superior a partir de Dados Abertos

Janaína A. Carvalho[1], Alison Rodrigo C. Souza[1], Rafael Gentil B. Santos[1], Ellen Polliana R. Souza[1], Marcelo Iury S. Oliveira[1]

[1]Universidade Federal Rural de Pernambuco (UFRPE)
Caixa Postal 063 – 56.900-000 – Serra Talhada – PE – Brasil

{janainaacarvalho, allison.rscavalcante, rafael.gentil, ellen.ramos, marcelo.iury}@ufrpe.br

Abstract. *Dropout in Higher Education is a subject discussed and defined by different authors in the literature, being characterized by the student leaving the course, the institution or the education system. In this sense, the objective of this work is the creation of an OLAP tool to aid in the decision making process on the avoidance rates in higher education, using open data from the Higher Education Census from 2009 to 2017, from the state of Pernambuco to validate the proposed solution. The results obtained bring the set of indicators raised at the end of this project, as well as the availability of the developed OLAP application, allowing to visualize the dropout in the state of Pernambuco.*
Keywords: *Data Mart, College Dropout, Open Data.*

Resumo. *A Evasão no Ensino Superior é um assunto discutido e definido por diferentes autores na literatura, sendo caracterizado pela saída do aluno do curso, da instituição ou do sistema de ensino. Nesse sentido, este trabalho tem como objetivo a criação de uma ferramenta OLAP para o auxílio no processo de tomada de decisão sobre as taxas de evasão no ensino superior, utilizando dados abertos do Censo da Educação Superior no período de 2009 a 2017, do estado de Pernambuco para validação da solução proposta. Os resultados obtidos trazem o conjunto de indicadores levantados ao final deste projeto, bem como a disponibilização da aplicação OLAP desenvolvida, permitindo visualizar a evasão no estado de Pernambuco.*
Palavras-chave: *Data Mart, Evasão Universitária, Dados Abertos.*

1. Introdução

O Ensino Superior Brasileiro teve uma expansão significativa ao longo dos anos 2000, a qual foi liderada pelo setor privado, seguida de uma expansão menor pelo setor público. Em 2007, foi implementado o Programa de Apoio a Planos de Reestruturação e Expansão das Universidades Federais (REUNI)[1] fortalecendo as Instituições de Ensino Superior (IES) públicas. Desse modo, o setor público conseguiu avançar no crescimento do número de matrículas, entretanto o número de concluintes não acompanhou o quantitativo de matrículas [Costa 2018], gerando um fenômeno social chamado evasão.

A evasão é um assunto discutido e definido por diferentes autores na literatura, sendo usualmente definida como a saída definitiva do aluno do sistema educacional, a

[1]Programa implementado a partir do Decreto Nº 6.096, de 24 de abril de 2007.

interrupção no ciclo de estudos [Gaioso 2005], saída do aluno da instituição de ensino antes de concluir o curso [Baggi and Lopes 2011] e/ou como a interrupção do aluno no ciclo do curso [Lüscher and Dore 2011].

Um estudo realizado pela comissão especial de estudos sobre a evasão nas universidades públicas brasileiras [ANDIFES et al. 1996], apresenta três indicadores de evasão, sendo eles: evasão de curso, caracterizado pelo desligamento apenas do curso sem desvínculo com a instituição; evasão da instituição, quando ocorre o desligamento da instituição na qual o aluno está matriculado; e evasão do sistema, na qual ocorre o abandono do ensino superior. Essa comissão afirma que a seleção do conceito mais adequado de evasão deve ser feito em função do objetivo pretendido no estudo a ser realizado.

Nesse estudo, o cálculo da evasão baseou-se tanto no conceito de evasão de curso, como no conceito de evasão de instituição, ambos seguindo as formas de cálculos listadas por [Silva and Lobo 2012], apresentadas na Tabela 3 e têm sido utilizadas em diversos trabalhos para cálculos de taxas anuais de evasão [Hoed 2016]. Nas fórmulas, M2 se refere a matrícula correspondente a um determinado ano, I2 é o número de ingressantes neste mesmo ano, M1 corresponde ao numero de matrículas no ano anterior e C1 é o número de concluintes no ano anterior. Para calcular a evasão na IES, é necessário descontar nos ingressantes o número de estudantes que mudaram de curso, mas não de IES (ITC2).

Tabela 1. Tipos de cálculo para a evasão.

Forma de Evasão	Cálculo
Evasão de Curso	1 – (M2 – I2)/(M1 – C1)
Evasão de IES	1- (M2 – I2 + ITC2)/(M1 – C1)

Com a implantação da Lei de Acesso a Informação (LAI) Lei 12.527/2011 [Brasil 2011], regulamentada pelo Decreto 7.724/2012 [Brasil 2012], tornou-se possível a verificação de dados públicos através da transparência ativa na qual o governo publica dados dos seus órgãos que sejam de interesse do cidadão [Veiga and Guimarães 2015]. Diante disso, a quantidade de dados disponibilizados tem crescido consideravelmente. Contudo, a forma bruta em que esses dados ainda são publicados, torna o processo de extração de informações relevantes e de apoio a decisão, pelo cidadão comum, dificultoso. Nesse contexto, a criação de inovações tecnológicas e o desenvolvimento de aplicações que facilitem analisar os dados abertos vem sendo impulsionada tanto pela academia como por gestores [Gonçalves et al. 2016].

Nesse sentido, este trabalho tem como objetivo analisar as taxas de evasão no ensino superior, utilizando dados abertos do Censo da Educação Superior no período de 2009 a 2017, do estado de Pernambuco para validação da solução proposta. Esta análise foi realizada através do desenvolvimento de uma ferramenta OLAP. Para tanto, busca-se responder as seguintes questões de pesquisa no presente estudo: **QP1 -** Quais indicadores são utilizados para analisar a evasão no ensino superior? **QP2 -** Qual status da evasão no estado de Pernambuco? **QP3 -** É possível analisar a evasão através dos dados abertos?

Este artigo está organizado da seguinte maneira: Na Seção 2, são apresentados os trabalhos relacionados. A Seção 3 apresenta o procedimento metodológico. A Seção 4 expõe os resultados e a Seção 5 apresenta as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Na literatura, alguns autores têm analisado quais indicadores estão relacionados a evasão, suas causas e seus impactos para a sociedade através da obtenção de dados extraídos a partir das próprias IES, como também através de dados abertos, geralmente disponibilizados pelo Instituto Nacional de Estudos e Pesquisas Educacionais (INEP)[2] e pelo Portal Brasileiro de Dados Abertos [3]. Após a obtenção dos dados, é realizada a preparação dos mesmos, selecionando as variáveis que serão analisadas, bem como padronizando-as, para que enfim sejam realizadas análises a partir de software estatísticos [Hoed 2016].

Por exemplo, [Santos 2017] propõe um modelo de aplicação de BI para monitoramento e combate da evasão escolar utilizando dados armazenados nos sistemas da instituição em estudo. Para desenvolver a aplicação, o autor seguiu as principais técnicas conhecidas: compreensão do negócio, compreensão dos dados, preparação dos dados, modelagem, avaliação e implementação. Ao final, foi possível mapear o aluno que evade por indicadores como: modalidade de ensino, nível, sexo, faixa etária e tipo da evasão.

[Orsi et al. 2016] também apresentam uma solução de visualização de dados acerca de ingressos e abandonos nos cursos da Universidade Federal do Estado do Rio de Janeiro (UNIRIO). O trabalho possibilitou a visualização de informações acerca da evasão por curso e por centro da instituição, por forma de ingresso e perfil geográfico.

Para analisar a evasão no ensino superior, são poucos os trabalhos disponíveis que fazem uso de aplicações interativas, adaptáveis e de alto poder analítico. Nesse sentido, o presente trabalho visa contribuir com a fomentação da importância dessas aplicações para tal análise, desenvolvendo um Sistema de Apoio à Decisão que servirá como auxílio para a análise da evasão no ensino superior.

3. Método

Para a análise proposta, foi desenvolvido um Data Mart baseado no modelo proposto por [Kimball 1996]. Os dados utilizados foram extraídos do Censo da Educação Superior. Para uma melhor análise de dados históricos, o período escolhido para o estudo foi de 2009 a 2017, filtrando as instituições e seus respectivos cursos localizados no estado de Pernambuco. Sendo utilizados os dados referentes as dimensões Aluno, IES e Cursos da base de dados do Censo. Além disso, foi necessário ainda, obter dados de outras fontes, como por exemplo, informações acerca da micro e mesorregião dos cursos e das instituições, extraídos a partir do servidor de dados do Instituto Brasileiro de Geografia e Estatística (IBGE)[4].

A área de negócio pertinente ao projeto foi especificada a partir do levantamento de indicadores necessários para a análise da evasão no ensino superior. Esta busca foi realizada, inicialmente, por meio de uma pesquisa bibliográfica, verificando nos trabalhos atuais, quais indicadores os mesmos utilizam e adaptando-os para este trabalho, além disso, foram realizadas entrevistas com gestores de instituições de ensino superior de Pernambuco, as quais tinham como objetivo extrair e validar indicadores importantes para o estudo da evasão.

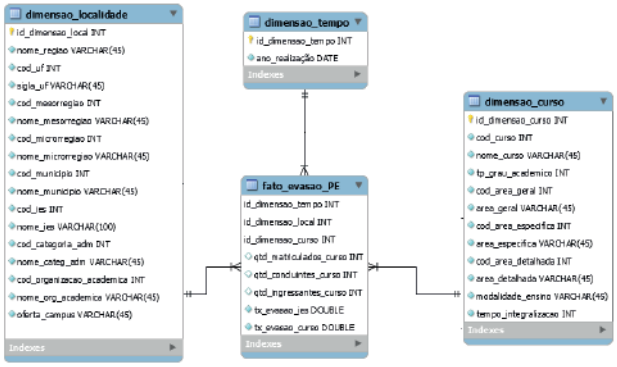
[2]http://www.inep.gov.br/

[3]http://dados.gov.br/

[4]ftp://ftp.ibge.gov.br/Pib_Municipios/2014/base/base_de_dados_2010_2014.xls

Para a criação do modelo dimensional, foi seguido o esquema estrela, caracterizado pela presença de dados redundantes, o que permite um melhor desempenho nas consultas. Sendo a tabela de fatos nomeada como "fato_evasao_PE" e as tabelas dimensões como "dimensao_tempo", "dimensao_localidade" e "dimensao_curso", conforme é apresentado na Figura 1, para esta etapa foi utilizada a ferramenta *MySQL Workbench*[5].

Figura 1. Modelo dimensional.



A criação do banco de dados ocorreu após a modelagem dimensional, sendo utilizado o SGBD relacional *PostgreSQL*[6]. Para a carga dos dados no banco, foi necessário realizar o processo de Extração, Transformação e Carga (ETC), no qual foi utilizada a ferramenta *Pentaho Data Integration*, integrante da plataforma *Pentaho Business Intelligence*. Com o *Data Mart* construído, foi então desenvolvida uma aplicação OLAP visando a oferta de um ambiente mais dinâmico e de fácil visualização. Para isto, foi utilizada a ferramenta *Microsoft Power BI*[7]. Esta ferramenta está disponível através do link *https://jncarvalho80.wixsite.com/website*.

4. Análise da Evasão

Nesta seção, são apresentados os resultados para as questões de pesquisa (QP) definidas na Seção 1.

4.1. QP 1: Quais indicadores são utilizados para analisar a evasão no ensino superior?

Após a revisão da literatura e análise dos trabalhos relacionados, foram identificados os indicadores utilizados por estes para analisar a evasão. Além disso, a partir das respostas das entrevistas aplicadas a gestores de instituições de ensino superior, novos indicadores foram identificados. O conjunto de indicadores formado ao final da pesquisa é apresentado na Tabela 2.

É válido ressaltar que, por ter sido realizada a pesquisa dentro de uma instituição de ensino, os indicadores levantados pelos respondentes foram referentes aos dados que a mesma dispõe. Entretanto, ao se trabalhar com dados abertos alguns desses indicadores não estão disponíveis, mesmo assim é possível obter informações relevantes e estratégicas para a área em estudo.

[5]https://www.mysql.com/products/workbench/
[6]https://www.postgresql.org/
[7]https://powerbi.microsoft.com/pt-br/

Tabela 2. Indicadores mapeados para análise da evasão.

Indicadores	
01 - Desempenho dos alunos com deficiência, para avaliar a ação afirmativa	10 - Distinção entre oriundos de escolas públicas e particulares e sua relação com a evasão
02 - Desempenho acadêmico	11 - Estudo da classe social dos alunos que mais evadem
03 - Renda familiar per capita	12 - Relação de alunos que entraram por cotas com a evasão
04 - Escolaridade dos pais	13 - Relaçao de alunos com bolsa de Iniciação Científica, monitorias e demais do genero, com a evasão
05 - Participação em atividades complementares (pesquisa e extensão)	14 - Quantitativo de alunos evadidos em número e percentual
06 - Custo por aluno	15 - Quantitativo de alunos evadidos por curso
07 - Evadido por área de conhecimento	16 - Quantitativo de alunos evadidos por ano
08 - Informações sobre Assistência Estudantil	17 - Quantitativo de alunos evadidos por gênero
09 - Quantidade de alunos evadidos por semestre	18 - Quantitativo de alunos evadidos por faixa etária

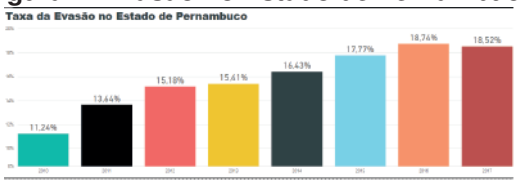
Para o presente trabalho, as visualizações possíveis são: evasão por curso, instituição, modalidade de ensino, categoria administrativa, organização acadêmica, ano, local em que a instituição está situada, quantitativo de alunos matriculados, concluintes e ingressantes.

4.2. QP 2: Qual status da evasão no estado de Pernambuco?

Após a construção do *Data Mart* foi desenvolvida uma aplicação OLAP, a qual permite a visualização de questões estratégicas relacionadas à evasão no estado de Pernambuco.

A Figura 2 apresenta a visualização da taxa de evasão no estado de Pernambuco, na qual é possível observar um crescimento contínuo moderado da evasão ao longo dos anos, com exceção de 2017, que mostrou-se estável comparando com o ano anterior. Para esta visualização foi utilizada a taxa de evasão calculada para instituições e sua evolução ao longo dos anos.

Figura 2. Evasão no Estado de Pernambuco.



Na Figura 3, são apresentados os quantitativos referente aos alunos matriculados, ingressantes, concluintes e evadidos. Nota-se um número expressivo para a quantidade de alunos que evadem por ano, possuindo uma significativa superioridade em relação aos alunos que concluem. Ressaltando que, nesse caso, foram considerados a quantidade de alunos que evadiram tanto de curso como de instituição, ou seja, mesmo que o aluno tenha saído do curso para ingressar em outro, na mesma IES, ele está sendo considerado nesta visualização.

Em relação ao tipo de categoria administrativa, apresentado na Figura 4, a categoria Privada sem Fins Lucrativos apresenta a maior média de evasão para o período estudado com 19,80%, seguida pela categoria Privada com Fins Lucrativos com 17,83%. No entanto, apesar da categoria Instituições Públicas apresentarem menores taxas, vale considerar que, quando somadas, apresentam uma elevada evasão. Isso não deixa de ser um fato preocupante, pois representa desperdício e prejuízo aos cofres públicos.

Figura 3. Quantidade de Alunos por Situação.

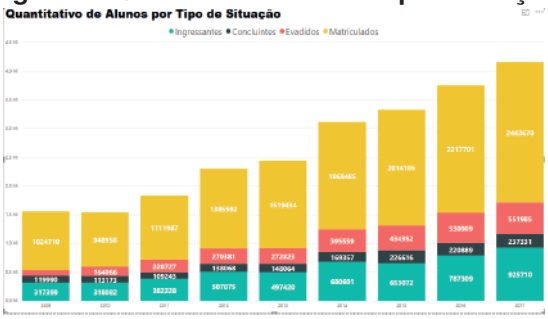
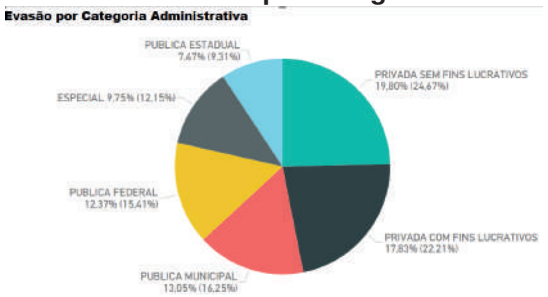


Figura 4. Média da Evasão por Categoria Administrativa.



No que tange ao tipo de Organização Acadêmica, a de maior média da taxa de evasão para o período foi por Centro Universitário com 24,29%. Esse resultado pode ser explicado devido a uma parte significativa destas organizações serem constituída por cursos ofertados à distância. Faculdades, Universidades e Institutos Federais possuem 15,36%, 16,52% e 17,41% de taxa de evasão, respectivamente.

No que se refere a modalidade de ensino, foi identificado que os cursos a distância apresentam uma média para a taxa da evasão, ao longo dos anos em estudo, de 21,98%, enquanto que a modalidade presencial apresenta como valor médio 13,50% da referida taxa. Esse número corrobora a literatura, a qual aponta os cursos de ensino à distância como aqueles que possuem maiores índices de abandono dos alunos.

Dado o exposto, diferentes análises podem ser realizadas acerca da Evasão no Ensino Superior. Desse modo, uma aplicação de BI pode contribuir no processo de análise da evasão, possibilitando aos gestores uma melhor compreensão dos dados disponibilizados pelas instituições, por permitir a análise em diferentes níveis de visualizações, comparando diferentes tipos de dados.

4.3. QP 3: É possível analisar a evasão através dos dados abertos?

Os dados abertos do Ensino Superior, disponibilizados a partir do Censo da Educação Superior, são importantes fontes de pesquisa sobre os estados das instituições, cursos, docentes e alunos do Ensino Superior. Em vista disso, para calcular a evasão, é necessário o quantitativo de alunos matriculados, ingressantes e concluintes do referido ensino, desse modo muitas análises podem ser realizadas através dos dados obtidos pelo Censo.

Como forma de validar os dados da aplicação disponibilizada, foi realizada uma comparação com os dados disponíveis nos relatórios de gestão da Pró-Reitoria de Planejamento e Desenvolvimento Institucional (PROPLAN) da UFRPE. Este relatório só está

disponível para os anos 2013, 2014 e 2017. Por isso, foram comparados apenas os valores correspondentes a estes anos, disponíveis na Tabela 3.

Tabela 3. Tabela comparativa das taxas de evasão.

		Ano		
	Fonte	**2013**	**2014**	**2017**
UFRPE	BI PCC	17,84%	21,27%	16,38%
	PROPLAN (UFRPE)	29,92%	24,92%	21%
UFRPE - UAST	BI PCC	14%	22%	17%
	PROPLAN (UFRPE)	30,75%	24,26%	22%
BSI - UAST	BI PCC	19%	24%	24%
	PROPLAN (UFRPE)	37,25%	26,98%	29,30%

Apesar de não ter sido encontrada uma relação entre os dados de 2013, foi possível observar que para 2014 os dados são bem próximos, além disso, entre 2014 e 2017 foi observada uma tendência de queda, para UFRPE e sua unidade UAST, tanto na aplicação apresentada como no relatório. Outro fator de explicação para essa diferença, é que o resultado do cálculo da evasão pode variar dependendo da forma de cálculo adotada [Lima and Zago 2018]. Neste trabalho, foi utilizado o cálculo segundo o Instituto Lobo [Silva and Lobo 2012].

Ao longo do desenvolvimento deste trabalho foram observadas algumas inconsistências na base de dados do Censo da Educação Superior, como por exemplo, no tempo de integralização do aluno alguns valores são disponibilizados em anos, outros em semestres, outros ainda possuem valores discrepantes e fora do padrão de disponibilização.

Diante da aplicação desenvolvida, mostra-se que é possível analisar a evasão a partir da utilização de dados abertos. Entretanto, as inconsistências encontradas nas bases disponibilizadas podem impactar na confiabilidade dos dados disponibilizados. Entre os problemas encontrados, destacam-se a falta de completude, qualidade de informação, valores em padrões diferentes, tornando-se inadequados para reúso, como mencionado por [Alcantara et al. 2015].

5. Conclusão e Trabalhos Futuros

Ao longo do desenvolvimento deste trabalho foi possível obter um conjunto de indicadores necessários para análise da evasão no ensino superior, os quais mostram a complexidade do estudo da evasão. Tendo como base os indicadores levantados, alguns foram adicionados a aplicação OLAP aqui desenvolvida, permitindo analisar a evasão sob diferentes aspectos, além disso foi possível corroborar o que a literatura menciona sobre os problemas encontrados ao se trabalhar com dados abertos.

Além dos dados aqui apresentados, este trabalho apresenta como contribuições a construção de um *Data Mart* voltado para o estudo da Evasão no Ensino Superior e a aplicação OLAP para apoiar os gestores no processo decisório da análise da Evasão. Como trabalhos futuros, pretende-se validar o *Data Mart* com o público alvo, para verificar se aplicação satisfaz a necessidade do mesmo. Além disso, é necessário ainda, incorporar ao *Data Mart* todos os indicadores levantados ao fim do projeto, sugeridos pelo público alvo, analisando as restrições e possibilidades dos mesmos. Pretende-se também fazer uso de mineração de dados, com o objetivo de identificar padrões e relações na base de dados montada ao fim do projeto.

Referências

Alcantara, W., Bandeira, J., Barbosa, A., Lima, A., Ávila, T., Bittencourt, I., and Isotani, S. (2015). Desafios no uso de dados abertos conectados na educaçao brasileira. In *Anais do Workshop de Desafios da Computação Aplicada à Educação. CSBC*.

ANDIFES, ABRUEM, and SESu/MEC (1996). Diplomação, retenção e evasão nos cursos de graduação em instituições de ensino superior públicas. avaliação. *Revista da Avaliação da Educação Superior*, pages 55–66.

Baggi, C. A. d. S. and Lopes, D. A. (2011). Evasão e avaliação institucional no ensino superior: uma discussão bibliográfica. *Avaliação: Revista da Avaliação da Educação Superior*, 16(2).

Brasil (2011). *LEI Nº 12.527, DE 18 DE NOVEMBRO DE 2011*.

Brasil (2012). *DECRETO Nº 7.724, DE 16 DE MAIO DE 2012*.

Costa, F. P. (2018). Acesso e Permanência no Ensino Superior: uma análise para as universidades federais brasileiras. *Universidade Federal de Pernambuco*.

Gaioso, N. P. d. L. (2005). *O fenômeno da evasão escolar na educação superior no Brasil. 2005. 75 f.* PhD thesis, Dissertação (Mestrado em Educação)–Programa de Pós-Graduação em Educação da Universidade Católica de Brasília.

Gonçalves, R., Viterbo, J., and Sousa, P. (2016). Um estudo preliminar sobre o uso de dados abertos na implementação de serviços para cidades inteligentes. In *II Workshop de Pesquisa e Desenvolvimento em Inteligência Artificial, Inteligência Coletiva e Ciência de Dados-2016-Niterói, RJ*, pages 122–131.

Hoed, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação. *Universidade de Brasília*, page 188.

Kimball, R. (1996). *The data warehouse toolkit: practical techniques for building dimensional data warehouses*. John Wiley New York.

Lima, F. S. and Zago, N. (2018). Desafios conceituais e tendências da evasão no ensino superior: a realidade de uma universidade comunitária. *Revista Internacional de Educação Superior*, 4(2):pages 366–386.

Lüscher, A. Z. and Dore, R. (2011). Política educacional no brasil: educação técnica e abandono escolar. *Revista Brasileira de Pós-Graduação*, 8(1).

Orsi, B. d. P., Góes, L. C., and Santoro, F. M. (2016). Desenvolvimento de indicadores para análise de desempenho e evasão de alunos da UNIRIO com a utilização de Self-Service BI. *Universidade Federal do Estado do Rio de Janeiro - UNIRIO*, page 56.

Santos, J. S. d. (2017). *Business Intelligence*: Uma Proposta Metodoógica para análise da Evasão Escolar em Instituições Federais de Ensino. *Universidade Federal do Paraná*.

Silva, Filho, R. L. L. e. and Lobo, M. B. d. C. M. (2012). Esclarecimentos metodológicos sobre os cálculos de evasão. *Instituto Lobo*, pages 1–7.

Veiga, J. C. d. A. and Guimarães, J. C. B. (2015). Análise de Dados Abertos Governamentais usando Técnicas de Business Intelligence: um Estudo de Caso das Eleições 2014. *PhD Proposal*, pages 1–67.

Uma Análise da Inclusão Escolar de Pessoas com Deficiência a partir de Dados Governamentais Abertos

Davi Maia[1], **Ellen Polliana R. Souza**[1], **Marcelo Iury S. Oliveira**[1]

[1]Universidade Federal Rural de Pernambuco (UFRPE)
Caixa Postal 063 – 56.900-000 – Serra Talhada – PE – Brasil

{davi.maia,ellen.ramos,marcelo.iury}@ufrpe.br

Abstract. *School inclusion is a process for insertion, maintenance and support of Persons with Disabilities (PwD) in regular teaching rooms. In this sense, this work aims to investigate school inclusion of PwD using. Such analysis was performed using a Data Mart developed through microdata of the School Census of Basic Education in the period from 2010 to 2018, with data from the state of Pernambuco to validate the proposed solution. Results show a positive evolution in enrollments of students with disabilities in regular education and their reduction in special education.*
Keywords: *Data Mart, School Inclusion, Open Data.*

Resumo. *Inclusão escolar é o processo de inserção, manutenção e apoio à Pessoas com Deficiência (PcD) em salas de ensino regular. Neste sentido, este trabalho tem como objetivo investigar aspectos da inclusão escolar de PcD. A análise foi realizada através de um Data Mart construído com os microdados do Censo Escolar da Educação Básica no período de 2010 à 2018, utilizando dados do estado de Pernambuco para validação da solução proposta. Resultados mostram evolução positiva nas matrículas de educandos com deficiência no ensino regular e redução das mesmas no ensino especial.*
Palavras-chave: *Data Mart, Inclusão Escolar, Dados Abertos.*

1. Introdução

Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE)[1], em seu último censo, há, no Brasil, 45,6 milhões de pessoas com algum tipo de deficiência, seja ela severa ou moderada. Dos vários tipos de deficiências ou limitações encontradas, este censo aponta a deficiência visual com o maior número de pessoas (35.791.488), seguida pela deficiência motora (13.273.969), auditiva (9.722.163) e, por fim, as deficiências mentais e intelectuais (2.617.025).

Os quantitativos dispostos acima apontam números significativos em relação a população com deficiência do Brasil. Esta população é assistida por uma série de políticas públicas [Brasil 2010] que buscam assegurar seus direitos e necessidades básicas.

Dentre os vários âmbitos destas políticas, certamente a educação ocupa um lugar estratégico. Até o início do século 21, o Sistema Educacional Brasileiro possuía dois tipos de serviços: a escola regular e a escola especial - ou o aluno frequentava uma, ou a outra [Alonso 2013]. Contudo, na última década, esse sistema modificou-se com

[1]http://www.portaldeacessibilidade.rs.gov.br/portal/index.php?id=noticias&cod=2128

a proposta inclusiva, resultando apenas na escola regular, que acolhe todos os alunos. Favorecendo, assim, a diversidade na medida em que considera que os alunos podem ter necessidades específicas em algum momento de sua vida escolar, além de transformar a escola em um espaço para todos.

Sabendo da importância da inclusão escolar de PcD, nos últimos anos intensificaram-se os estudos com a preocupação de analisar e diagnosticar a realidade social e escolar de PcD. Esses estudos são de fundamental importância na discussão e articulação de novos modelos sociais e políticos, com objetivo de reduzir, ou mesmo eliminar, o caráter seletivo e discriminatório da educação.

Entretanto, muitos dos estudos encontrados na literatura se concentram em um período específico de tempo ou em uma determinada localidade. Desta forma, essas análises não podem ser facilmente utilizadas por gestores e a população em geral a tomarem melhores decisões e desenvolverem projetos voltados para as suas localidades. Nesse contexto, os dados abertos podem ser importantes aliados para a realização de estudos sobre o processo inclusivo, uma vez que atualmente são publicadas quantidades massivas de dados a respeito da educação no Brasil. Dados abertos contribuem, assim, para a transparência, para o desenvolvimento de soluções e para a tomada de decisão.

Os dados abertos educacionais do Brasil são publicados em formatos que dificultam sua descrição e reutilização, oferecendo muito esforço e custo de processamento para se gerar informações [Alcantara et al. 2015]. Para análise da inclusão escolar, os dados utilizados como base também são os dados educacionais, que além das dificuldades apresentadas, devido ao formato, também apresentam problemas quanto a legibilidade e qualidade dos mesmos.

Nesse sentido, este trabalho tem como objetivo apresentar uma análise sobre a inclusão escolar de PcD, utilizando microdados do Censo Escolar da Educação Básica no período de 2010 à 2018, do estado de Pernambuco para validação da solução proposta. Esta análise foi desenvolvida através de um Data Mart que pode ser reutilizado para análise da inclusão escolar para outras localidades e para quaisquer períodos de tempo. Para tanto, este trabalho buscou responder às seguintes questões de pesquisa: **QP1 -** Quais indicadores podem ser utilizados para analisar a inclusão escolar de PcD?; **QP2 -** Como se dá a inclusão escolar de PcD no Estado de Pernambuco?; **QP3 -** É possível analisar a inclusão escolar através dos dados abertos?

Este artigo segue organizado da seguinte forma: na Seção 2, são apresentados os trabalhos relacionados, bem como as contribuições destes para o presente estudo. Na Seção 3, é definido o método utilizado neste artigo. Na Seção 4, são apresentados os experimentos realizados. A Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Alguns trabalhos encontrados na literatura realizam análises da inclusão escolar de PcD a partir de dados abertos governamentais. Entre eles, destacamos os trabalhos de [Gomes 2010], [Caiado 2011], [Meletti and Ribeiro 2014] e [Meletti 2014] buscam, primariamente, realizar a extração de dados a partir de alguma fonte externa, normalmente dados governamentais abertos.

E preciso destacar que até onde nossa revisão alcançou, não foram encontrados

nenhum trabalho que propusesse algum tipo de modelagem analítica de dados. De uma forma geral, as atividades de extração, limpeza, padronização e agrupamento dos dados foram realizados de forma manual. Nos quais as tabelas eram extraídas a partir de sua fonte externa, os dados eram limpos e padronizados evitando quaisquer inconsistências e sendo agrupados segundo a necessidade da pesquisa. Além disso, toda informações construída somente foi divulgada através de trabalhos técnico-científicos, o que dificulta o acesso da população de um modo geral.

Apesar de oferecem informações importantes, os trabalhos relacionados não trazem um panorama atual da realidade da inclusão escolar, uma vez que tem dados muito desatualizados a respeito desse processo. Sobre a disposição das análises, os estudos trazem análises de visualização em sua maioria em forma de tabelas e com indicadores ligados a área estatística, os quais demandam um determinado conhecimento do público para a investigação dos mesmos, oferecendo dificuldades para o público alvo destas.

O presente trabalho complementa os trabalhos relacionados ao desenvolver uma análise do processo inclusivo através de um Sistema de Apoio à Decisão. Desta forma, aprimorando os processos de extração, transformação e carga dos dados, melhorando o processo de disponibilização da informação, sendo esta de forma mais simples para qualquer tipo de usuário.

3. Método

Para a realização das análises, foi desenvolvido um Data Mart, seguindo a metodologia de desenvolvimento apresentada por [Palestino 2001, Kimball and Ross 1996]. Esta metodologia compreende oito etapas desde o planejamento até a implantação do sistema. A abordagem escolhida para a construção do Data Mart foi a *bottom-up*.

Os dados necessários para a criação do Data Mart foram extraídos do Portal Brasileiro de Dados Abertos[2] do período de 2010 à 2018 e as informações territoriais de cidades e municípios do estado de Pernambuco, extraídas da Portal do IBGE[3]. Os microdados do Censo Escolar da Educação Básica estão disponíveis no formato CSV, os dados apresentam cerca de 92 campos divididos em 3 grandes áreas: informações básicas a cerca do aluno como idade, se tem deficiência, entre outros; informações da escola em que o educando está inserido; e informações sobre a etapa de ensino na qual ele estuda, como o tipo de sala, a série. As bases de dados do Censo Escolar contém um dicionário de dados para cada tabela, os quais definem os dados contidos nas mesmas, bem como seus tipos e representações. Os dados territoriais apresentam campos divididos em 2 grandes áreas, uma relativa ao território com nome e códigos das cidades, microrregiões e estados, e outra relativa ao PIB dos municípios que para este trabalho será descartada.

Foi criado um modelo dimensional *Star-Schemma* no qual foi definida uma tabela Fato, nomeada como "Fato-Matriculas", e seis Dimensões, chamadas de "dim_tempo", "dim_local", "dim_sexo", "dim_faixa_etaria", "dim_modalidade", "dim_etapa_ensino"como apresenta a Figura 1.

Após a criação do banco de dados, a partir do modelo dimensional, os dados foram carregados através do processo de Extração, Transformação e Carga (ETL), por

[2]http://dados.gov.br/dataset/microdados-do-censo-escolar

[3]https://ww2.ibge.gov.br/home/estatistica/economia/pibmunicipios/2010_2013/default_base.shtm

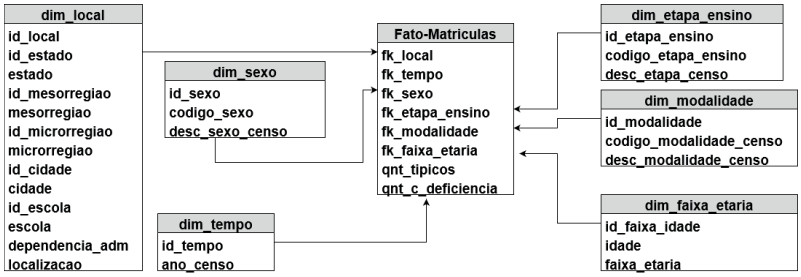


Figura 1. Modelo Dimensional. Fonte: Elaborado pelos autores

meio da ferramenta *Pentaho Data Integration*. Nessa etapa, os dados foram extraídos, limpos, filtrados, agrupados e carregados no SGBD relacional *PostgreSQL*. Essas etapas envolvem a escolha do formato da fonte de dados original, os campos que serão utilizados para o estudo, a padronização de dados com formatação diferente, retirada de caracteres especiais, e filtragem utilizando a presença de algum dado específico. Após a realização do processo de ETL, foi utilizada a ferramenta *Microsoft Power BI*, como ferramenta OLAP - *Online Analytical Processing*, para realização da consulta dos indicadores propostos no modelo dimensional. Esta ferramenta está disponível através do link *http://davymaia3.wixsite.com/mapa-da-inclusao*.

4. Análise da Inclusão de PcD

Nesta Seção, são apresentados os resultados para as questões de pesquisa definidas anteriormente na Seção 1.

4.1. Quais indicadores podem ser utilizados para analisar a inclusão escolar de PcD?

Após a revisão da literatura, conforme apresentado na Tabela **??**, foram identificados os indicadores utilizadas na análise da inclusão escolar de PcD. Além da revisão, também foi realizada uma entrevista estruturada com gestores responsáveis pela educação inclusiva na cidade de Serra Talhada, os quais definiram, por grau de relevância, os indicadores que consideravam importantes para análise e avaliação desse processo.

A Tabela 1 apresenta os indicadores mapeados. Como resultado, além dos indicadores de evolução de matrículas, são apresentados indicadores relacionados à informações geográfica do aluno e da instituição, à situação escolar (modalidade e etapa de ensino), condições de permanência na instituição e acessibilidade, tais como ofertas e salas de AEE, banheiros e dependências adaptados.

Como pode ser observado na Tabela **??**, os indicadores de análise temporal das matrículas de educandos com deficiência e de modalidade de ensino do educando são analisados por todos os autores, devido sua importância este indicador também será analisado neste trabalho. A análise de matrículas por estado/microrregião/cidade foi realizada por todos os trabalhos exceto nos de [Meletti and Bueno 2011] e [Meletti and Ribeiro 2014], uma vez que estes realizaram estudos no contexto do país. Este indicador também será analisado neste presente trabalho. O agrupamento de zonas rurais e urbana foi utilizado apenas no trabalho de [Caiado 2011] e não será analisado neste trabalho. A modalidade e a etapa de ensino são aspectos para analisar a inserção de alunos com deficiência nas

Tabela 1. Indicadores mapeados para anlise da incluso escolar

Indicador	
1 - Evolução de matrículas de educandos típicos e com deficiência	6 - Matrículas de Educandos típicos e com deficiência por etapa de ensino
2 - Matrículas de Educandos típicos e com deficiência por faixa etária	7 - Matrículas de alunos típicos e com deficiência por rede de ensino
3 - Matrículas de Educandos típicos e com deficiência por localização	8 - Instituições de ensino com salas de AEE
4 - Matrículas de Educandos típicos e com deficiência por sexo	9 - Instituições de ensino com ofertas do recurso AEE
5 - Matrículas de Educandos típicos e com deficiência por modalidade de ensino	10 - Instituições de ensino com dependências e banheiros adaptados para PcD

salas regulares e em salas especiais e de Educação de Jovens e Adultos (EJA) e devido a sua importância também será levado em consideração neste trabalho. O sexo e a faixa etária dos educandos é importante como forma de analisar políticas inclusivas para determinados grupos e também para verificar a distorção Idade-Série para esta população, respectivamente, e será também analisado neste trabalho.

4.2. QP2 - Como se dá a inclusão escolar de PcD no Estado de Pernambuco?

A Figura 2 apresenta a evolução das matrículas de alunos típicos e com deficiência no período de 2010 à 2018 no estado de Pernambuco. Através do gráfico, é possível verificar um aumento significativo na matrícula de alunos com deficiência, o qual pode estar relacionado às legislações sancionadas para esta população, nesse período, permitindo um maior acesso destes ao ensino base em Pernambuco.

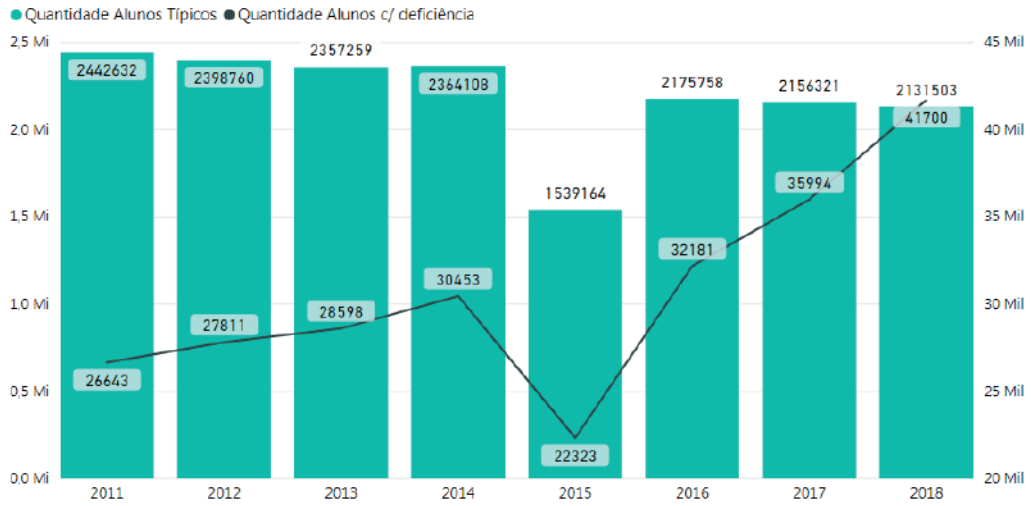


Figura 2. Evolução de matrículas de educandos típicos e com deficiência. Fonte: Elaborada pelos autores

A Figura 3 apresenta a evolução das matrículas de alunos com deficiência por modalidade de ensino. É possível verificar que o ensino regular é o detentor do maior quantitativo de matrículas, ao mesmo tempo que é a modalidade de ensino que apresentou maior crescimento ao longo dos anos, crescendo mais de 100% ao longo dos dez anos analisados. A EJA apresentou crescimento significativo ao longo dos anos, aumentando cerca de 100% do seu total, refletindo na preocupação de escolarização de PcD que tem distorção Idade-Série. O ensino especial sofreu redução na quantidade de matrículas passando a deter menos de 50% das matrículas, em comparação com o ano de 2010.

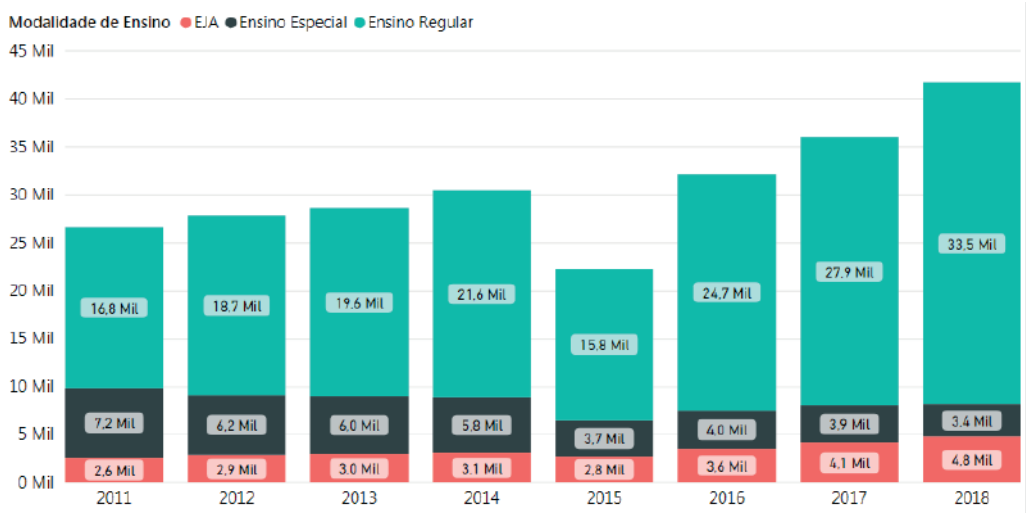


Figura 3. Evolução de matrículas de educandos com deficiência por modalidade de ensino. Fonte: Elaborada pelos autores

A Figura 4 apresenta as matrículas de alunos com deficiência por etapa de ensino. Observa-se que o Ensino Fundamental é a etapa de ensino que mais detém matrículas de PcD, tendo crescimento considerável no total. O ensino infantil, por sua vez, apresentou queda no primeiro ano de análise, retomando o crescimento nos anos seguintes. O ensino médio teve um significativo crescimento no contexto geral, refletindo uma melhor escolarização de PcD. A EJA teve bom crescimento ao longo dos anos.

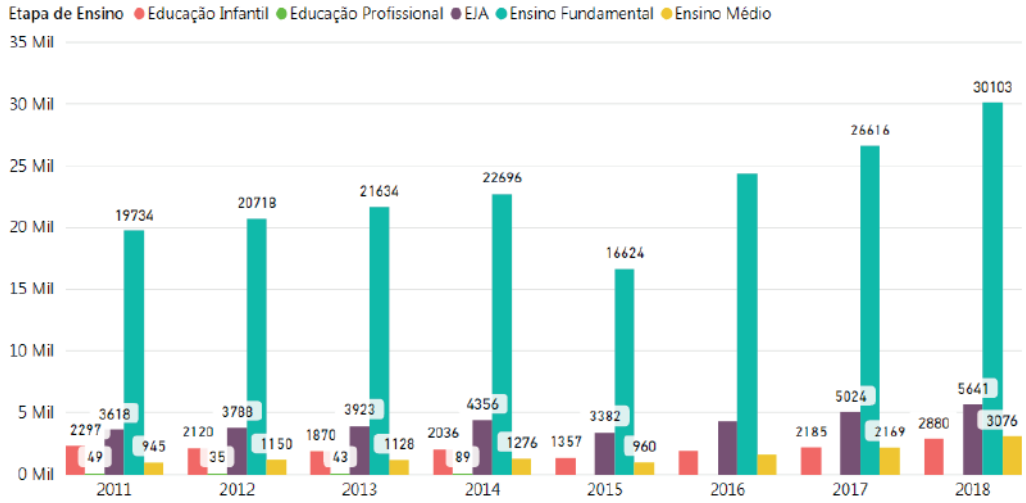


Figura 4. Evolução de matrículas de educandos com deficiência por etapa de ensino. Fonte: Elaborada pelos autores

A Figura 5 apresenta as matrículas de alunos com deficiência por faixa etária, sendo a faixa de 7 a 14 anos como aquela que contém mais alunos matriculados, mostrando que grande parte destes encontram-se no ensino fundamental. É possível verificar também que essa faixa etária não apresentou evolução durante os anos analisados. Algumas faixas etárias que apresentaram evoluções. Mais especificamente, as faixas de 4 a 6 anos e a de 15 a 17 anos evidenciam um crescimento de matrículas na pré-escola e no

ensino médio.

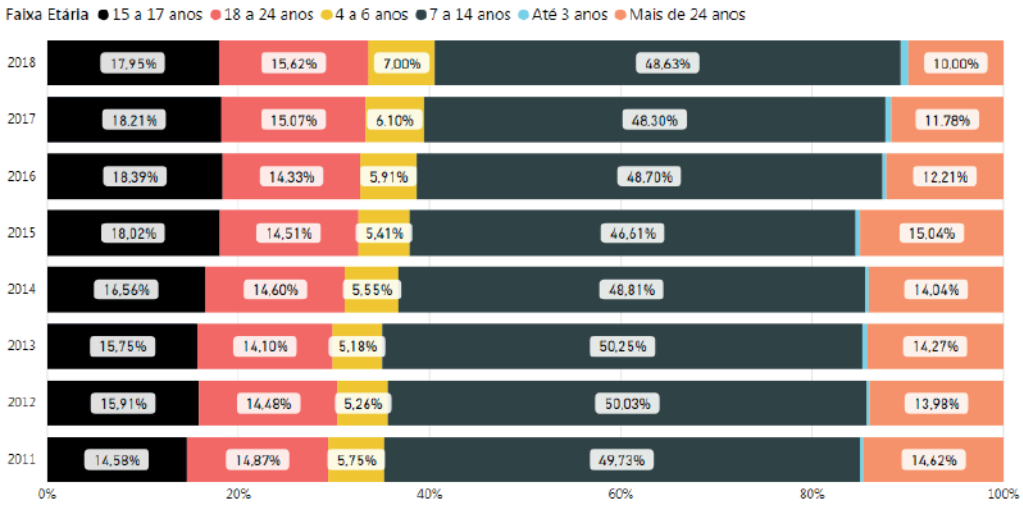


Figura 5. Evolução de matrículas de educandos com deficiência por faixa etária. Fonte: Elaborada pelos autores

4.3. QP3 - É possível analisar a inclusão escolar através dos dados abertos?

Os dados abertos do Censo Escolar servem como base para análise sobre o processo de inserção de educandos com deficiência em instituições de ensino. Através dos mesmos, é possível caracterizar os educandos a partir de sua modalidade e etapa de ensino, se os mesmos recebem Atendimento Educacional Especializado, e se as escolas tem dependências adaptadas para estes.

Para análise do processo inclusivo, se fazem necessários investigações sobre currículos, atividades, avaliação de aprendizagem, e professores com formação adequadas para PcD [Mantoan 2003]. Esses estudos precisam não somente analisar números ou dados, mas também situações sociais da realidade de cada educador [Ferreira Baptista et al. 2018].

Entretanto, analisar a inclusão escolar somente a partir de dados abertos é uma tarefa ainda complexa, uma vez que estes apresentam falta de completude, qualidade da informação e formatos de dados inadequados para reúso, como relata [Alcantara et al. 2015]. Um interessante caso ocorre em uma escola na cidade de Serra Talhada que tem matrículas de educandos em sua sala de ensino especial no ano de 2016, mas de acordo com os dados apresentados nas consultas OLAP, tais matrículas não existem. Isso reflete problemas com relação a quem informa esses dados ao sistema de coleta do Censo Escolar, bem como a validação desses dados junto as instituições.

5. Conclusão e Trabalhos Futuros

Neste artigo, foi apresentada uma análise da inclusão escolar de PcD a partir de Dados Abertos. Até onde sabe-se, este é o primeiro trabalho a criar um Sistema de Apoio à Decisão para analisar a inclusão escolar de PcD, bem como o primeiro a disponibilizar tais informações de forma mais clara e objetiva aos usuários.

Através das análises, identificaram-se 10 indicadores que podem fornecer importantes informações para análise do processo inclusivo. Além disso, foi possível identificar

que houve uma evolução positiva na matrículas de educandos com deficiência o que mostra maior preocupação no acesso deste à escola. Também foi possível verificar aumento de matrículas de educandos com deficiências em salas regulares e nos ensinos médio e EJA, mostrando preocupação com a integração destes educandos na escola regular, bem como na escolarização de educandos com distorção idade-série.

Apesar de uteis, os dados abertos apresentam uma série de limitações. Entre elas, destacam-se a falta de completude, a legibilidade, a qualidade da informação e o formato dos dados. Como trabalho futuro, pretende-se expandir o modelo dimensional, de forma que este atenda indicadores educacionais de cunho social como as taxas de escolarização, alfabetização e distorção Idade-Série. Também pretende analisar os quantitativos relacionados as salas de AEE e aos dados de PcD em instituições de Ensino superior.

Referências

Alcantara, W., Bandeira, J., Barbosa, A., Lima, A., Ávila, T., Bittencourt, I., and Isotani, S. (2015). Desafios no uso de dados abertos conectados na educaçao brasileira. In *Anais do DesafiE-4o Workshop de Desafios da Computação Aplicada à Educação. CSBC*.

Alonso, D. (2013). Os desafios da educação inclusiva: foco nas redes de apoio. *InNova-escola. São Paulo: Abril*.

Brasil (2010). Cartilha do censo pessoas com deficiência. *Secretaria de Direitos Humanos da Presidência da República (SDH/PR)/Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência (SNPD)*.

Caiado, K. R. M. (2011). Educação especial na educação do campo: 20 anos de silêncio no gt 15. *Revista Brasileira de Educação Especial*.

Ferreira Baptista, V., Guimarães Cardoso, M. V., and Lobato Martins, U. (2018). A inclusão das pessoas com deficiência no sistema educacional como instrumento viabilizador da igualdade: exposição e análise crítica dos respectivos indicadores. *Capital Científico*, 16(1).

Gomes, C. G. S. (2010). Escolarização inclusiva de alunos com autismo na rede municipal de ensino de belo horizonte. *Revista Brasileira de Educação Especial*.

Kimball, R. and Ross, M. (1996). *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons.

Mantoan, M. T. E. (2003). *Inclusão Escolar: O que é? Por quê? Como fazer*. Moderna, São Paulo.

Meletti, S. M. F. (2014). Indicadores educacionais sobre a educação especial no brasil e no paraná. *Revista Educação e Realidade*.

Meletti, S. M. F. and Bueno, J. G. S. (2011). Educação infantil e educação especial: Uma análise dos indicadores educacionais brasileiros. *Revista Contrapontos*.

Meletti, S. M. F. and Ribeiro, K. (2014). Indicadores educacionais sobre a educação especial no brasil. *Cadernos Cedes*.

Palestino, C. B. (2001). *BI-business intelligence: modelagem e tecnologia*. Axcel Books, Rio de Janeiro.

Sistema *Web Crawler* para Coleta Automática de *Tweets*, Persistência em Bancos de Dados e Análises Estatísticas

Andrea Mourelo Rodriguez[1], Patrick S. Sava[2], Arthur Ituassu[3], Sérgio Lifschitz[1]

[1]Departamento de Informática – (PUC-Rio) - Rio de Janeiro - RJ

[2]Top Free Games (TFG) – São Paulo - SP

[3]Departamento de Comunicação – (PUC-Rio) - Rio de Janeiro - RJ

andrea.mourelo@student.ecp.fr, patrick.sava@gmail.com,
ituassu@puc-rio.br, sergio@inf.puc-rio.br

Abstract. *This work presents a web crawler system designed to extract historical data from Twitter, a very large data streaming social network. The main challenge involves dealing with both online data streams and old, offline, data, automatically fetched as web-based pages updated by scrolling. These data is persisted in a relational database and, once cleaned-up and formatted, we may use them as samples for statistical and sentimental analysis.*

Resumo. *Este trabalho apresenta um sistema de navegação na web para extração de dados históricos do Twitter, uma rede social com publicação contínua de grandes volumes de dados. O principal desafio consiste em lidar com dados online offline, que devem ser capturados automaticamente e atualizados no estilo de páginas web, por meio da opção de scrolling. Os dados são persistidos em um banco de dados relacional e, após limpeza e formatação, podemos usá-los como amostras para análises estatísticas e de sentimentos.*

1. Introdução

Com o uso ativo da internet pela população de todo o planeta, as redes sociais se tornaram espaços repletos de informação sobre tudo quanto é assunto. Esses dados espalhados pela internet, sobretudo dentro das redes sociais, constituem uma massa de dados de alto valor para análises sentimentais e quantitativas. Neste trabalho, o foco é dado sobre a rede social *Twitter*[1]. Trata-se de um serviço compartilhado onde as mensagens que circulam contêm, de maneira geral, muita informação com poucos caracteres.

A rede social digital *Twitter* surgiu em 2006 [Farhi 2009] com a proposta de ser um *microblog* interpessoal em massa cujos usuários poderiam se comunicar por meio de mensagens curtas chamadas de *tweets*, de até no máximo 140 caracteres (280, desde 2017). Diversos estudos já foram feitos baseados nas mensagens que circulam dentro dessa rede social e tiveram muitos resultados promissores. Por meio de análises dos conteúdos das mensagens, já foi possível, entre outros, prever resultados de eleições políticas [Lovejoy 2012] e realizar análises de saúde pública [Paul 2011].

1 https://www.twitter.com/

Este trabalho envolve o estudo de alternativas existentes para coletar mensagens antigas escritas por usuários da rede social *Twitter* e filtrar esses dados por período e assunto, gerando amostragens de mensagens possíveis de serem usadas em futuros estudos por terceiros. Para isso foi desenvolvido um sistema de banco de dados para permitir que o usuário escolha um assunto, um período de tempo e, em seguida, possa de maneira transparente e sem conhecimentos técnicos, extrair os dados já publicados que se enquadram nas definições da busca diretamente da base do *Twitter*.

O principal desafio científico e tecnológico, que justificou a implementação deste sistema, envolveu a extração de *tweets* "antigos", que não estão disponíveis pela maneira convencional que a rede social provê. Este trabalho também discute desafios e contribuições envolvendo a gestão e processamento de grandes volumes de *tweets*, desde a modelagem conceitual até o estudo de estruturas persistentes e mais adequadas. Inicialmente, as pesquisas começaram como um trabalho de iniciação científica, depois evoluiu para um protótipo associado a um projeto final de conclusão de curso de graduação [Sava 2016] e atualmente se encontra disponível na web (http://tc.biobd.inf.puc-rio.br/) com uso controlado para fins acadêmicos.

2. Contexto e Motivação

O *Twitter* dispõe de uma API (*Application Programming Interface*) que disponibiliza para o público os *tweets* de maneira totalmente estruturada e com semântica conhecida por usar o padrão JSON[2]. Construída baseada na arquitetura REST [Fielding 2000], essa API permite a integração da rede social em diversas aplicações externas. Utilizada em larga escala, ela é extremamente eficiente mas tem uma grande limitação. Devido à enorme quantidade de *tweets* inseridos diariamente, para manter o desempenho deste serviço esses dados ficam disponíveis somente por um pequeno período de tempo. Essa limitação não causa nenhum problema em grande parte das aplicações já que estas estão interessadas em coletas em tempo real, mas restringe o acesso aos *tweets* passados.

O acesso a esses dados que já deixaram de ser indexados pela API do Twitter é difícil. A única forma que o próprio *Twitter* provê de consultas históricas é através da web pela chamada Busca Avançada[3]. Lá é possível escolher diversos filtros como as *hashtags*, período, localização do usuário quando do envio de *tweets*, entre outros.

Os resultados são exibidos no site apenas para visualização. Dessa forma, torna-se ruim para processamento e avaliação de dados em massa. Isso porque os dados não estão estruturados para se extrair as informações específicas, como conteúdo de cada *tweet*, o nome do usuário que realizou a postagem, em qual data, entre outras informações relevantes. Esta foi uma das principais motivações deste trabalho.

3. Trabalhos Relacionados

No seu blog oficial[4], o Laboratório de Estudos sobre Imagem e Cibercultura (LABIC) da Universidade Federal do Espirito Santo (UFES) publicou um estudo em que eles dizem ter capturado *tweets* antigos através de um software solicitado por seus pesquisadores.

2 https://tools.ietf.org/html/rfc4627
3 https://twitter.com/search-advanced
4 http://www.labic.net/blog/buscas-retroativas-no-twitter-fazendo-redes/

Através deste, eles dizem que sempre tiveram interesse em realizar essas buscas retroativas no *Twitter*. O propósito deste projeto é o mesmo do software que foi solicitado e usado por eles para coletar os *tweets*. Não é divulgada nenhuma informação a respeito do software utilizado pelo LABIC, portanto, não pode ser feito uma comparação entre o programa deles e o produto deste trabalho.

Um produto que está no mercado que tem um propósito semelhante é o *PYLON for Facebook Topic Data* criado pela DataSift[5]. Esse produto analisa e exibe informações relevantes sobre o que tem sido comentado na rede social Facebook[6] em relação a uma empresa ou produto para que seus clientes possam estudar o resultado de uma campanha na rede social ou simplesmente ver o que as pessoas falam sobre eles. Também da DataSift, tem o *Open Data Processing for Twitter*[7], esse produto faz analises parecidas com o *PYLON for Facebook Topic*, mas utilizando a Twitter API como forma de obtenção dos comentários. Nenhum destes dois produtos se preocupa com o que já foi dito no passado, ambos tratam somente o que tem sido escrito a partir do momento em que são configurados para funcionarem.

O projeto TubeKit[8] que é uma ferramenta para extrair diversas informações da rede social de vídeos YouTube[9]. Através dele é possível coletar links para vídeos, comentários e até informações de perfil de usuários da rede com possibilidade de filtrar o conteúdo por alguns atributos.

Alguns outros projetos interessantes também foram encontrados durante essa pesquisa e são brevemente mencionados e descritos a seguir:

- **BackTweets**[10] - permite que qualquer pessoa faça buscas por *tweets* antigos contendo uma palavra, hashtag ou link. É interessante para pessoas que querem ver os *tweets* que contém algum link ou que fazem referência a uma determinada palavra.
- **Snapbird**[11] - permite que qualquer pessoa veja *tweets* antigos de uma determinada pessoa, seguindo alguns outros filtros relacionados ao próprio usuário que está fazendo a busca
- **TwimeMachine**[12] - permite ver *tweets* antigos do próprio usuário apenas.

Em resumo, ao menos publicamente, a melhor forma de ter acesso a *tweets* antigos com restrição temporal segue sendo pela busca avançada do *Twitter*. Entretanto, conforme já dito anteriormente, ela não é boa para coletar dados em massa. Inclusive essa necessidade de ter acesso a *tweets* de forma retroativa já foi levantada pelos pesquisadores do LABIC-UFES. Este nosso trabalho, portanto, é válido pois proporciona uma maneira de obter esses dados de maneira alternativa à API do *Twitter*, com menos limitações, de maneira estruturada e com semântica integrada.

5 http://www.datasift.com/
6 https://www.facebook.com/
7 http://datasift.com/products/open-data-processing-for-twitter/
8 http://www.tubekit.org/
9 http://www.youtube.com/
10 http://backtweets.com/
11 https://snapbird.org/
12 http://www.twimemachine.com/

4. Implementação

Nosso projeto é composto por uma aplicação Web e uma aplicação *standalone*. Cabe observar que esta opção traz interesse adicional porque faz com que a complexidade de implementação do produto como um todo aumente, já que tem que se garantir que ambos funcionem em conjunto mesmo que sejam totalmente desacoplados. A aplicação Web funciona como interface de entrada e saída de dados para a aplicação *standalone* que trata de todo o processamento em *back-end* para o usuário. Não existe uma comunicação direta entre as duas aplicações porque não havia necessidade para isso. Definimos que o meio por onde elas se comunicam, mesmo que indiretamente, é via banco de dados.

O servidor utilizado para hospedagem da aplicação web, crawler e também do banco de dados é uma instância de servidor Linux que está situado no BioBD - Laboratório de Bioinformática e Banco de Dados da PUC-Rio. Eventualmente o serviço poderá ser disponibilizado em provedores de serviço na nuvem públicos, caso os requisitos de desempenho e volume de dados aumentem consideravelmente.

4.1. Crawler

Para a implementação do *web crawler*, foi escolhida a linguagem *Python*. A escolha foi baseada na familiaridade com a linguagem, por ser uma linguagem com possibilidades de programação estruturada ou orientada a objetos e porque foram encontradas várias boas bibliotecas de terceiros para auxílio na implementação das funções principais.

Esse programa foi construído para poder ser executado em paralelo com outras instâncias dele próprio. A motivação principal por trás disso é poder reduzir o tempo necessário para se fazer a pesquisa completa. Como a data de início da busca e data de final são dados de entrada de cada consulta, o modelo de dados foi pensado para que o resultado correspondente ao período de tempo de entrada se tornasse a união do resultado de cada um dos dias pertencentes ao intervalo completo. Como os resultados da página de busca avançada do *Twitter* não dependem dos dias anteriores, cada dia pode ser executado de forma totalmente independente. Portanto, esse modelo permite que cada dia do intervalo seja processado em paralelo, localmente ou em máquinas distintas. A execução de cada instância do *crawler* é feita automaticamente por meio do *Crontab*[13] presente nos sistemas operacionais derivados do Unix.

A principal dificuldade deste trabalho, e que o torna mais do que um "simples" *web crawler,* vem da maneira em que a página de onde se extraem os dados foi implementada. A página de resultados da busca avançada do Twitter, por poder retornar inúmeros resultados, implementa uma mecânica de *scroll to update*. Isso é muito comum em redes sociais e outros sistemas que precisam mostrar muito conteúdo. Com essa mecânica o cliente pode solicitar uma parte do conteúdo e, quando o usuário se aproxima do fim, o cliente solicita mais conteúdo e os concatena no final da página. Esse modo de operação economiza banda de comunicação de dados com respeito ao que seria necessário para transmitir toda a informação de uma só vez para o cliente. Como transmitimos um trecho pequeno por vez, essa comunicação torna-se muito mais rápida.

13 http://man7.org/linux/man-pages/man5/crontab.5.html

Um *web crawler* simples faz uma requisição HTTP(S), espera a resposta do servidor no formato HTML e extrai desse código HTML todas as informações desejadas. Por vezes nem todas as informações desejadas estão presentes na resposta do servidor e isso se torna um problema, que correspondeu ao impasse mais complexo na implementação desse programa. A solução que implementamos consiste em utilizar uma ferramenta de simulação de navegador para manter e poder executar os *scrolls* na página HTML até um determinado ponto e só depois recolher todo o código obtido para extração dos dados.

O objetivo da nossa ferramenta é coletar amostras de *tweets* e não todos os *tweets* escritos no determinado período de tempo porque para coletar todos seria muito custoso. Com base na teoria da amostragem [Mood 1950], foi decidido e implementado dois pontos de parada para o processo de *scroll* do *crawler*. O primeiro, e mais natural, é quando terminam os *tweets* daquele dia e o segundo é um limite de scrolls interno no código que passa da etapa de *scrolling* para a etapa de extração das informações. Esse limite pode ser configurado somente por quem dá manutenção ao sistema, o usuário não tem poder de mudar esse valor. Atualmente ele é suficiente para gerar amostras de tamanho bem considerável para amostragem.

Como simulador de navegador (*browser*) foi escolhido o *Selenium Web Driver* por ser compatível com a linguagem Python e por ser *open-source*. Ele trata de carregar as páginas HTML, extrair o código-fonte para processamento com o *Beautiful Soup* e também tem a função de "rolar" a página e forçar o carregamento de mais *tweets*.

O *virtual frame buffer* foi necessário porque o *Selenium* exige que exista um monitor para abrir e exibir uma instância de algum navegador previamente instalado na máquina, neste caso o Mozilla Firefox[14], e como o *crawler* é uma aplicação que será executada numa máquina servidora sem monitor, era preciso simular um. Para isso foi usado a aplicação *Xvfb(X virtual frame buffer)*[15] que faz essa função e tem um *wrapper* para Python, conhecido como *PyVirtualDisplay*[16]. E, por fim, a biblioteca *Beautiful Soup* disponível para *Python* foi a escolhida para fazer o processamento (*HTML parsing)* do código HTML e coletar as informações importantes.

4.2. Interface Web

O sistema web para agendamento e gestão das pesquisas solicitadas, é o intermediador entre o usuário e o *crawler* da *web* que a princípio não terão interação direta. Esse sistema foi pensado para seguir a arquitetura *Model View Controller* [Krasner 1988] na construção da aplicação web. O *front-end* dessa aplicação é bem simples, utilizou-se apenas HTML, CSS e um pouco de *JavaScript* para criar todas as telas onde o usuário interage com o sistema. Para poder usar o sistema, os usuários devem se cadastrar primeiro e ativar a conta por meio de um link recebido por e-mail. Os administradores do sistema devem aceitar o usuário e, a partir desse momento, ele pode fazer buscas.

Na Figura **1**, temos uma tela de resultado de consulta. Neste caso, o assunto buscado foi "'sergio moro' OR moro" no período de 01/06/2019 até 30/06/2019, contabilizando

14 https://www.mozilla.org/pt-BR/firefox/new/
15 https://www.x.org/archive/X11R7.6/doc/man/man1/Xvfb.1.xhtml
16 https://pypi.python.org/pypi/PyVirtualDisplay

55896 *tweets* no período de quatro semanas. Esse assunto permite coletar os *tweets* tendo a palavra ‘moro’ ou as duas palavras ‘sergio’ e ‘moro’ juntas. Além desses números, o usuário tem a possibilidade de fazer o download completo dos resultados em formato CSV. Esse formato é útil porque facilita a leitura por outros softwares e permite a visualização por diversos softwares de planilha. Isso permite aproveitar todo o potencial desses softwares para realizar análises sobre os dados.

Figura 1 – Resumo de uma busca específica

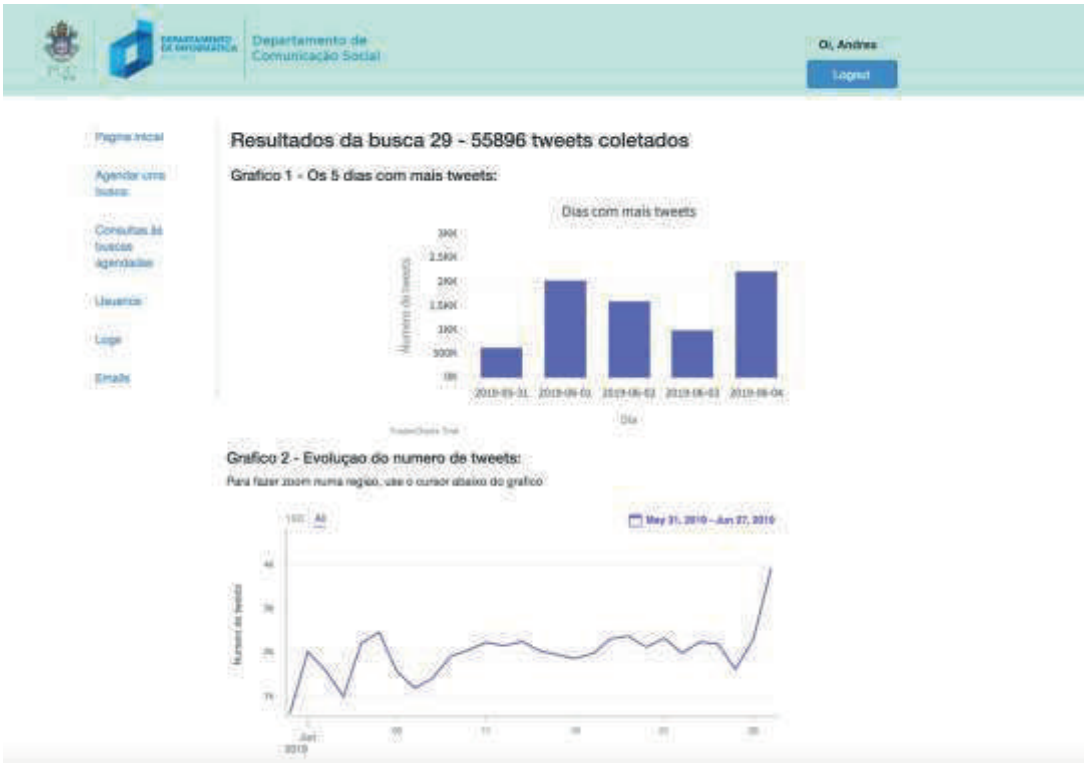


Figura 2. Tela inicial de análise dos resultados

Também, como demonstra a Figura 2, e em função das permissões do usuário, este pode visualizar algumas métricas e gráficos simples criados a partir da biblioteca *FusionCharts*, é também efetuar uma análise da mídia presente nos *tweets* (URLs).

Como teste da ferramenta, foi feita uma pesquisa buscando pelo assunto "Bolsonaro", no período de 01/05/2019 até 22/05/2019, para observar as menções ao sobrenome do Presidente do Brasil. Nós incluímos este assunto, com ou sem *hashtag*, no nome do usuário, nos textos do *tweet* e nos usuários referenciados por *replies* ao *tweet*.

Na Figura 3 vemos o gráfico exibido pela interface gráfica que mostra a distribuição de *tweets* por dia de consulta e também algumas tabelas dando informações sobre os resultados. Podemos notar que o número de *tweets* é constante e que as pessoas que mais criam conteúdo com essa palavra não são sempre mídias.

Tabela 1 - Numero de tweets feitos por midia:

MIDIA	NUMERO DE TWEETS
@DiarioPE	3
@JornalOGlobo	7
@OGloboPolitica	2

Tabela 2 - Ocorrências do assunto (Bolsonaro):

NUMERO DE OCORRÊNCIAS	RESULTADO
Numero medio por tweet (com #)	0.029130515906335
Numero medio por tweet	0.7812529153839

Tabela 3 - 10 pessoas que mais compartilharam:

USERNAME	NUMERO DE TWEETS
@HideoKojima	71
@ArwenLuthien_	58
@valeriosobral	45
@tarlouze	36
@OSul_noticias	35
@Marcelo68736229	35
@MariaProvesi	34
@PedroMottaB	34
@Duarte_Cau	33
@aEduardocGomes	31

Figura 3. Algumas métricas e visualizações sobre o conjunto de *tweets*

Na **Figura 4**, vemos uma das tabelas permitindo analisar a mídia compartilhada pelos usuários nos *tweets*: observamos muitas redes sociais e sites de mídias tradicionais. A quantidade total de *tweets* coletados foi 42.876 para o período de aproximadamente 3 semanas, lembrando que a ferramenta limita a quantidade de *tweets* por dia. É importante ressaltar que todos esses resultados estão distribuídos dentre os dias do intervalo de busca.

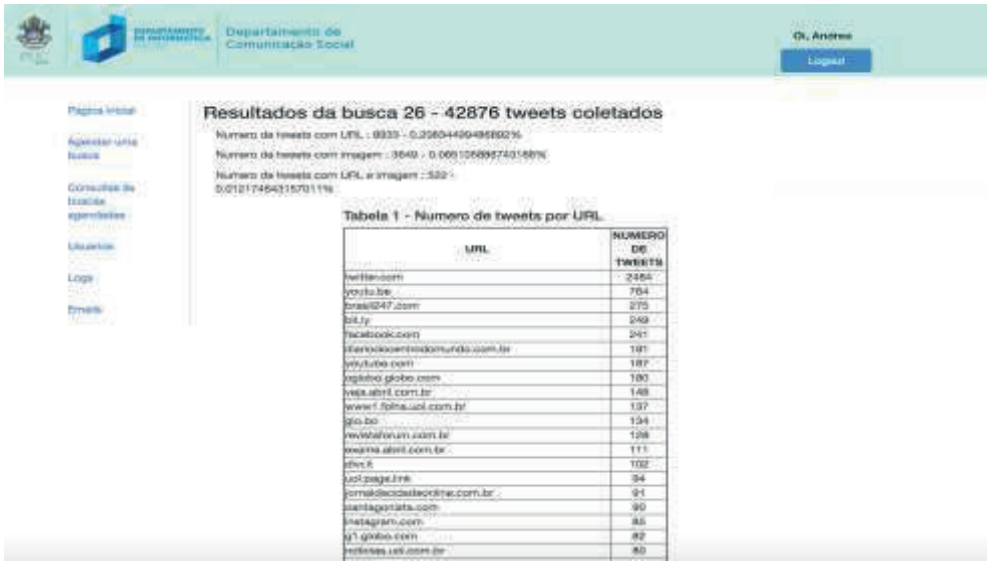


Figura 4. Análise da mídia compartilhada nos *tweets*

5. Conclusões

Este trabalho descreve uma ferramenta para extrair *tweets* antigos sem usar a Twitter API. A comunidade acadêmica e científica que pesquisa sobre dados de redes sociais pode usar a ferramenta para coletar seus dados. O produto já foi usado em diversas publicações [*e.g.*, Ituassu 2015, Capone 2017].

Como possíveis trabalhos futuros, pensamos em incluir na ferramenta uma funcionalidade de análise de sentimentos, e outros programas para processar os dados extraídos e viabilizar novas análises. Além disso, também poderia ser desenvolvida uma ferramenta de processamento de streaming do *Twitter* para complementar as análises.

Referências

Capone, L., Ituassu, A., Lifschitz, S. e Mannheimer, V. (2017). *Superposters, especialização e serviço: a Primeira Consulta Pública do Marco Civil da Internet no Twitter*. Revista Fronteiras vol. 19, p. 263-276.

Ituassu, A. e Lifschitz, S. (2015). *Temas e mídia em# Eleições2014: Twitter, opinião pública e comunicação política no contexto eleitoral brasileiro*, E-compós.

Fielding, Roy T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*; Doctoral dissertation, University of California, Irvine.

Lovejoy, Kristen; Waters, Richard D.; Saxton e Gregory D. (2012). *Engaging stakeholders through Twitter: How nonprofit organizations are getting more out of 140 characters or less*. Public Relations Review, vol. 38, no. 2, p. 313-318.

Mood, Alexander McFarlane (1950). *Introduction to the Theory of Statistics*.

Sava, P. (2016). *Web Crawler de Informações do Twitter para Persistência em Banco de Dados*, Projeto Final de Graduação, Departamento de Informática, PUC-Rio.

Paul, Michael J.; Dredze, Mark (2011). *You are what you Tweet: Analyzing Twitter for public health*. ICWSM vol 20, p. 265-272.

BioBD-ENEM: Migrando Grandes Planilhas para um Sistema de Banco de Dados na Web

Alexandre W. Vieira, Gabriel Cantergiani, Mariana D.A. Salgueiro, Stefano V. Pereira, Victor Augusto L.L. de Souza, Rafael P. de Oliveira, Sérgio Lifschitz

[1]Departamento de Informática
PUC-Rio, Rio de Janeiro - RJ

{e1512647, e1320681, c1510988, e1611082}@grad.inf.puc-rio.br

victoraugustolls@gmail.com,{rpoliveira,sergio}@inf.puc-rio.br

Abstract. *This work presents the design, development and implementation of the BioBD-ENEM System. The main challenges involve dealing with large and complex spreadsheets to import from the INEP (Education Ministry), information conceptual modeling, relational database implementation and fine tuning activities, besides a set of specialized graphics for data analysis.*

Resumo. *Este trabalho apresenta o projeto, desenvolvimento e implementação do Sistema BioBD-ENEM. Os desafios envolvem desde o tratamento e importação de planilhas complexas e volumosas da base de dados do INEP (Ministério da Educação), passando pela modelagem conceitual das informações, implementação e sintonia fina em um banco de dados relacional, além da análise dos dados através de gráficos especializados.*

1. Introdução

Uma das provas mais importantes da educação brasileira é o Exame Nacional do Ensino Médio (ENEM), ministrado pelo Instituto Nacional de Estudo e Pesquisa (INEP) [INEP 2019]. A prova objetiva avaliar a qualidade do ensino médio no país e é utilizada por diversas universidades como parte do processo seletivo dos cursos de graduação.

O INEP publica os resultados detalhados das provas, possibilitando que escolas e interessados avaliem os dados e extraiam informações úteis sobre o desempenho dos alunos e instituições. Estes resultados são disponibilizados em planilhas formato CSV [Society 2019]. Os arquivos seguem o seguinte modelo: cada linha se refere a um candidato que realizou a prova e cada coluna informa desde um número de identificação dos candidatos até as respostas para o questionário socioeconômico. Como a cada ano milhões de candidatos se inscrevem para o ENEM, a planilha com resultados fica imensa. Isso torna a manipulação dos dados no formato bruto complicada. Editores, como Microsoft Excel, tem dificuldades em abrir e visualizar arquivos da ordem de 7GB de dados.

Este trabalho apresenta o projeto de integração de dados e visualização realizado pelo laboratório BioBD da Pontíficia Universidade Católica do Rio de Janeiro (PUC-Rio). Apresentamos uma ferramenta capaz de consultar os dados com resultados das provas do ENEM de diversos anos. Houve um trabalho grande de modelagem de banco de dados, sintonia fina (*tuning*) e estudo de visualização de grandes volumes com alto desempenho.

Durante o trabalho foi necessário normalizar os diferentes formatos de planilhas disponibilizados pelo INEP e integrar as informações em um único banco de dados. Uma ferramenta web de consulta foi desenvolvida para manipular os dados e um servidor de aplicação configurado para rodar a aplicação. Utilizamos, como documento de referência para visualização, os gráficos propostos por Jorge Frias em seu manuscrito de dissertação de mestrado profissional do Departamento de Matemática da PUC-Rio [de Frias 2015].

2. Projeto e Modelagem do Banco de Dados

Ao acessar o site do INEP, é necessário escolher qual ano do ENEM quer-se baixar. O arquivo compactado contém: os dados das provas, um dicionário de dados, um ”leia-me”e outros documentos técnicos. Neste conjunto são recolhidos dados sobre as habilidades e competências de cada prova, além das provas em si e seus gabaritos. Cada arquivo possui entre 3GB e 7GB, dependendo do ano. As planilhas são muito volumosas e torna-se inviável manuseá-las diretamente.

Existe o problema de abrir a planilha em editores como o MS Excel, já que esta só lida com até pouco mais de 1 milhão de linhas e há pelo menos 5 milhões de participantes realizando a prova por ano. As planilhas, em formato CSV, têm em média 150 colunas, dificultando também a compreensão pelo usuário comum. A solução sugerida foi realizar um processo de ETL (*Extract-Transform-Load*), através de um script em Python, para manusear os dados em um SGBD relacional, no caso, o SBGD PostgreSQL. Devido às características dos dados, não houve dúvida quanto à opção pelo modelo relacional.

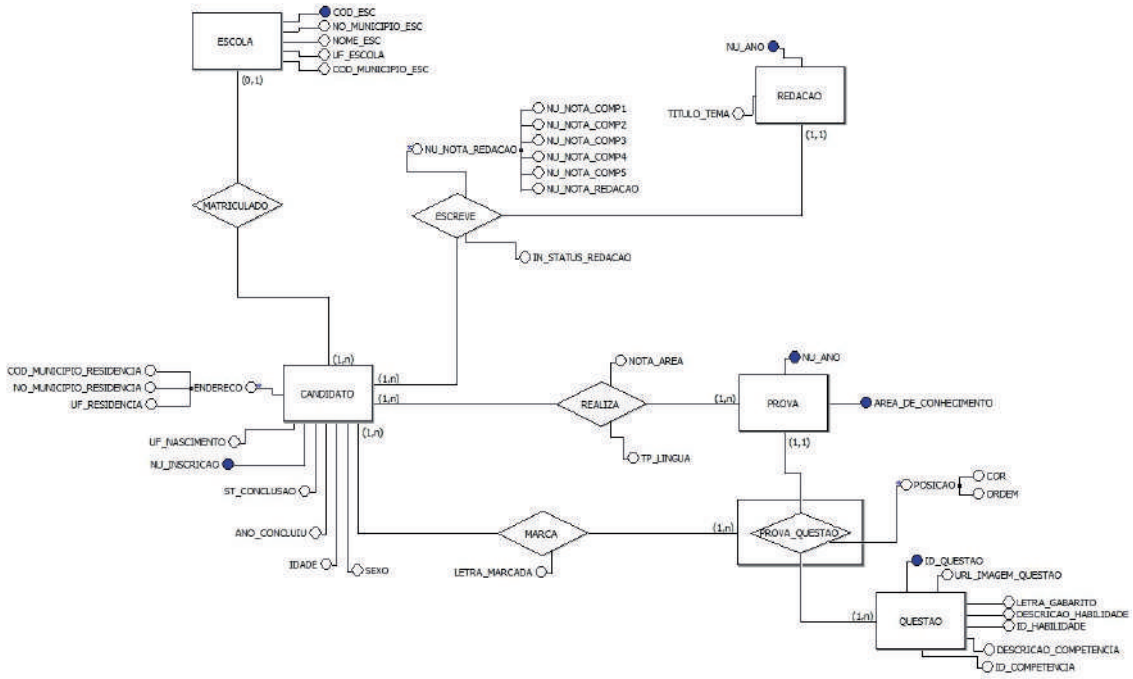


Figura 1. Diagrama de Entidades e Relacionamentos (DER)

Uma etapa inicial, e fundamental, foi realizar a modelagem conceitual de forma a permitir a compreensão dos dados presentes nas planilhas. Optamos por gerar um Diagrama de Entidades e Relacionamentos (DER) que representasse de forma abstrata e integrada os dados de diferentes arquivos CSV para os diferentes anos.

Após diversas aproximações e versões, o diagrama atual é ilustrado na Figura 1. Já na Figura 2 temos o DER sem atributos, facilitando sua compreensão com foco nas entidades relevantes. Em seguida, na Figura 3, temos um detalhamento de uma parte importante da abstração das planilhas originais, envolvendo candidatos e questões.

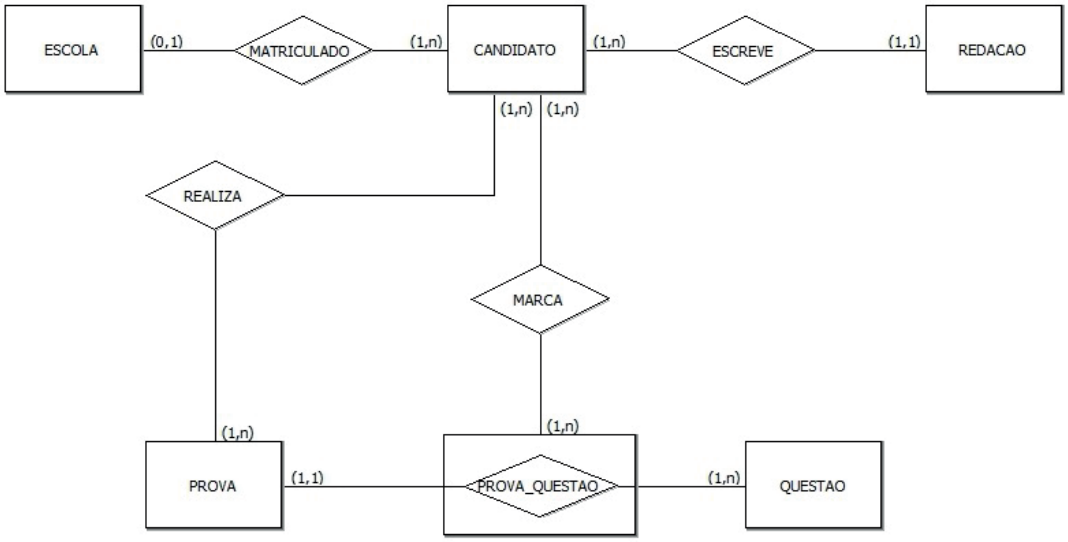


Figura 2. DER Sem Atributos

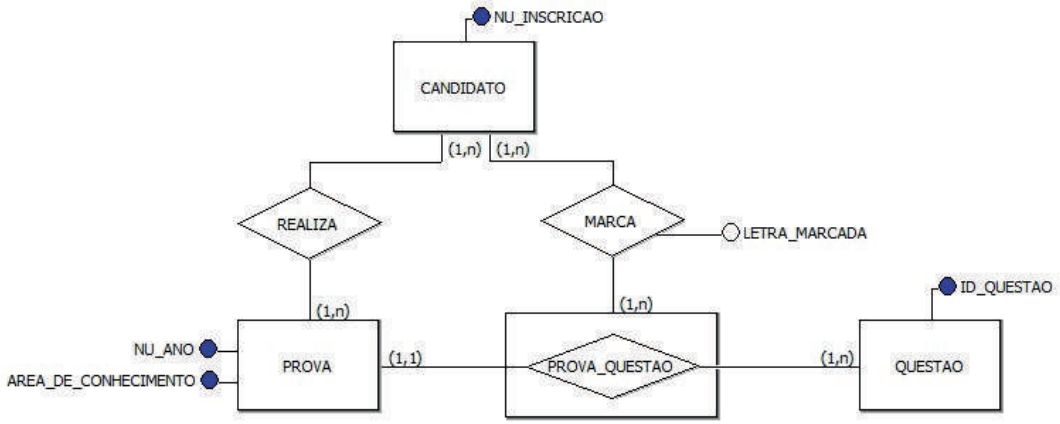


Figura 3. Relacionamento Candidato-Prova-Questão

A partir do diagrama entidade relacionamento, foi possível desenvolvermos um mapeamento para o banco relacional. O Sistema Gerenciador de Banco de Dados (SGBD) considerado foi o PostgreSQL em sua versão 9.7.

As seguintes tabelas foram criadas a partir do ER: *Microdados, Escola, Candidato, Redação, Prova, Realiza, Questão e Marca*. Com as tabelas criadas, a fase seguinte consistiu da importação das planilhas CSV através de scripts Python. Foram encontrados alguns problemas no processo de ETL pois as planilhas de cada ano não mantêm um padrão e podem se diferenciar na maneira como representam as informações. Por exemplo, uma nota nula é representada ora por '.', ora por espaço em branco.

Também há a dificuldade de entender a numeração das questões já que variam por cor de prova. Uma mesma prova é apresentada em versões diferentes. Por exemplo, no ano de 2012, a 1a questão da prova azul corresponde à 15a da prova amarela, 34a na prova rosa e 11a na prova branca. Assim, adotamos uma das cores (azul, no caso) como base e o gabarito das demais provas foi convertido para a numeração desta prova base. Logo, só precisamos armazenar uma solução por questão - diferença que chamamos em nossa modelagem por questões ”lógicas”(conceitual) versus questões ”físicas”(numeradas).

Após a padronização dos dados na tabela Microdados, eles foram redistribuídos nas tabelas correspondentes. Fizemos a transferência dos dados da tabela de Microdados para as outras tabelas citadas. Para realizar a redistribuição, também foram utilizados scripts Python. Vários atributos necessitaram de tratamentos e ajustes antes de serem transferidos, visando padronizar por ano.

Um dos tratamentos realizados pelo script Python é a generalização da ordem das questões e das respostas de cada estudante. No caso dos microdados, as respostas de cada um(a) estão uma seguida da outra em uma cadeia de caracteres (*string*) e estão associadas à prova realizada, referindo-se à cor da prova e, portanto, à sua ordem. Através do tratamento dos scripts e após a inserção dos dados já tratados na tabela Marca, sabe-se quais questões cada estudante acertou, não importando a ordem das questões da prova.

3. Implementação: sintonia fina e threads

Para lidar com o grande volume de dados após a integração das provas, foi necessário aplicar técnicas de sintonia fina (tuning) de banco de dados para acelerar a execução de consultas. Um dos grandes desafios encontrados na replicação dos gráficos da dissertação de mestrado profissional em [de Frias 2015], e da criação da ferramenta web em si, foi a demora na qual os mesmos eram gerados [de Souza 2018].

O sistema inicial não possuía nenhum tipo de índice que poderia ajudar na execução das consultas para geração dos gráficos. Por esse motivo, foi realizada uma pesquisa de todos os tipos de índices oferecidos pelo PostgreSQL com o objetivo de encontrar o mais adequado para cada situação. Para a grande maioria dos casos, o modelo *hash* apresentava o melhor resultado teórico para os casos de uso, entretanto, a documentação do SGBD recomendava não utilizar o mesmo em versões anteriores à versão 10, dada sua instabilidade e inconsistência. Por esse motivo, para as situações encontradas, o índice padrão em *Árvore B+* foi considerado a melhor opção a ser utilizada.

Além da criação de índices, buscou-se também otimizar o uso do espaço de disco ocupado pelo banco, visando reduzir o número de páginas acessadas e obter um melhor desempenho. Para isso, foi necessário compreender exatamente o que poderia ser retirado de útil dos dados disponibilizados, com a finalidade se encontrar melhores modelagens que não prejudicassem a extração das informações, tornando-as objetivas o suficiente para retirar o que fosse desejado. A compreensão do modelo de avaliação do ENEM, das necessidades das escolas e das limitações técnicas existentes, permitiu o desenvolvimento de mudanças pontuais em boa parte da modelagem física, gerando uma redução considerável do espaço ocupado em disco e do tempo de acesso aos dados.

Para algumas consultas apenas a criação de índices não foi suficiente. Como os dados do ENEM não sofrem alteração, a criação de visões materializadas permite que os

resultados das consultas mais lentas sejam pré-calculados e armazenados para um acesso mais rápido. Foi também elaborado um estudo para compreensão do formato de armazenamento de uma página e de uma tupla pelo PostgreSQL. Foi possível economizar espaço em disco através do uso de uma estratégia chamada *Column Tetris*, que consiste na ordenação das colunas de uma tabela de forma que o espaço ocupado seja mínimo.

Um dos principais problemas referentes à experiência do usuário que utilizava a aplicação Web era a demora no carregamento dos gráficos. Cada vez que um gráfico era selecionado, uma consulta complexaera realizada no Banco de Dados, demorando até 10 minutos. Além disso, os resultados não eram armazenados: para voltar a um gráfico anterior, era necessário refazer a consulta. Para solucionar este problema implementamos um sistema de *threads* para processar cada gráfico separadamente. Para isso, foi necessário reestruturar toda a organização de como os gráficos eram gerados, separando as lógicas de programação e consultas ao banco em funções independentes. Quando uma escola é selecionada, são criadas diferentes *threads* (uma para cada gráfico disponível), que passam a processar as consultas em paralelo, guardando os resultados em arquivos temporários para uso futuro. A grande vantagem desta abordagem é otimizar o tempo em que o usuário analisa um gráfico para já carregar e processar os próximos que serão vistos. Além disso, guarda-se um resultado já visto caso o usuário queira voltar para um gráfico anterior.

4. Visualização por Gráficos

A ferramenta oferece uma análise de gráficos gerais, na qual é possível ter uma visão geral de um estado ou mesmo do Brasil inteiro. Há também uma seleção de escolas, onde o interessado pode fazer uma consulta para uma escola escolhida, uma comparação entre escolas e uma análise da evolução, seja do estado ou da escola desejada, ao longo dos anos. Alguns exemplos de gráficos em nosso sistema são mostrados a seguir.

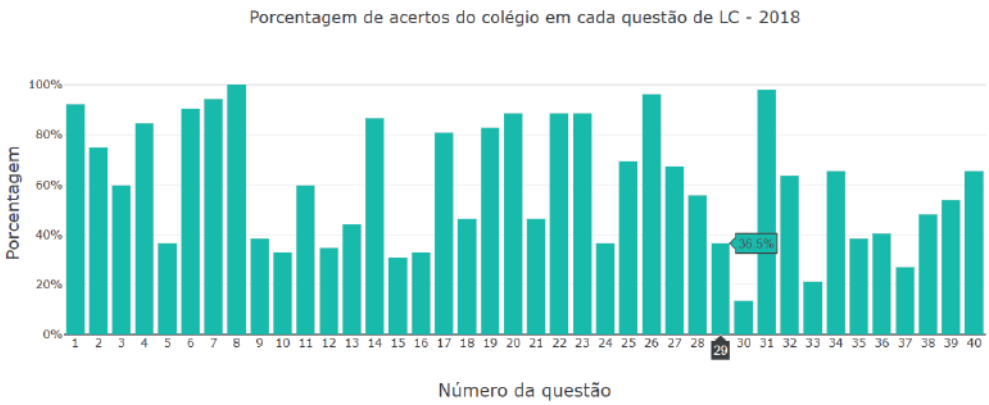


Figura 4. Porcentagem de Acertos por Questão

O gráfico de porcentagem de acertos por questão [Fig. 4] mostra para cada questão de uma área da prova a porcentagem de acerto da escola escolhida. Como referência é utilizada a prova azul para estabelecer a ordem das questões, mas todas as provas normais contêm as mesmas questões. Com esse gráfico e a disponibilidade de ver a prova em PDF ao apertar um botão, é possível saber quais são as questões exatas que estudantes da escola escolhida mais acertaram e mais erraram. Dessa forma, os professores podem trabalhar em cima de questões parecidas.

Já o gráfico de divisão dos candidatos por escola (Fig 5 mostra a comparação ano a ano entre os candidatos matriculados em alguma escola e os não vinculados a nenhuma escola. Já é possível notar uma curiosidade: aqueles que não estão vinculados a nenhuma escola são maioria dentre os candidatos que realizam a prova do ENEM.

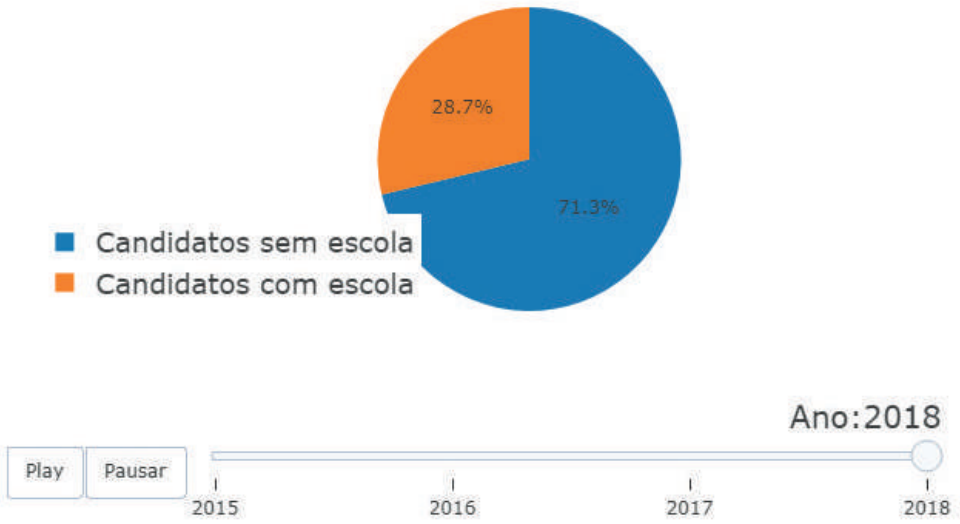


Figura 5. Divisão dos Candidatos por Escola

O gráfico de distribuição de alunos por acerto (Fig. 6 mostra a quantidade de alunos que obtiveram um certo número de acertos nas provas de cada área de conhecimento. Neste exemplo, na prova de Ciências Humanas e suas Tecnologias, a maioria dos alunos do Brasil acertou 11 questões, não por acaso próximo de 20% (5 opções).

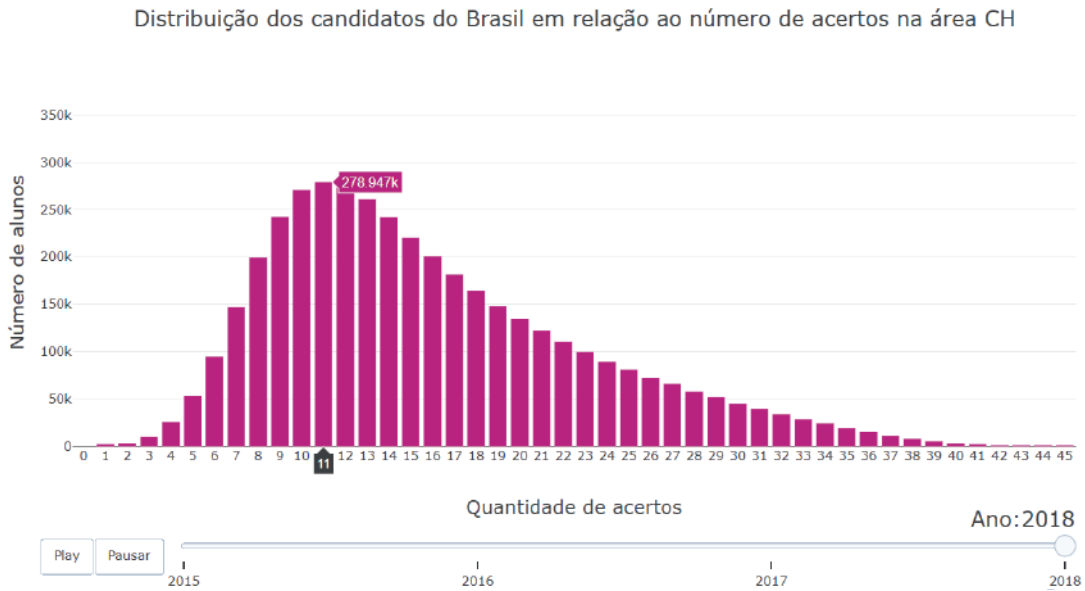


Figura 6. Distribuição de Alunos por Acerto

A Figura 7 é voltada especificamente para a redação, pois são analisadas as competências desta prova. Nele, é possível ver o número de alunos da escola escolhida em função da nota recebida. Nesse caso, pode-se fazer uma comparação do nível de excelência pra cada competência e, dependendo da nota de cada um, os professores podem estabelecer um nível de preferência para treinar essas competências nos estudantes.

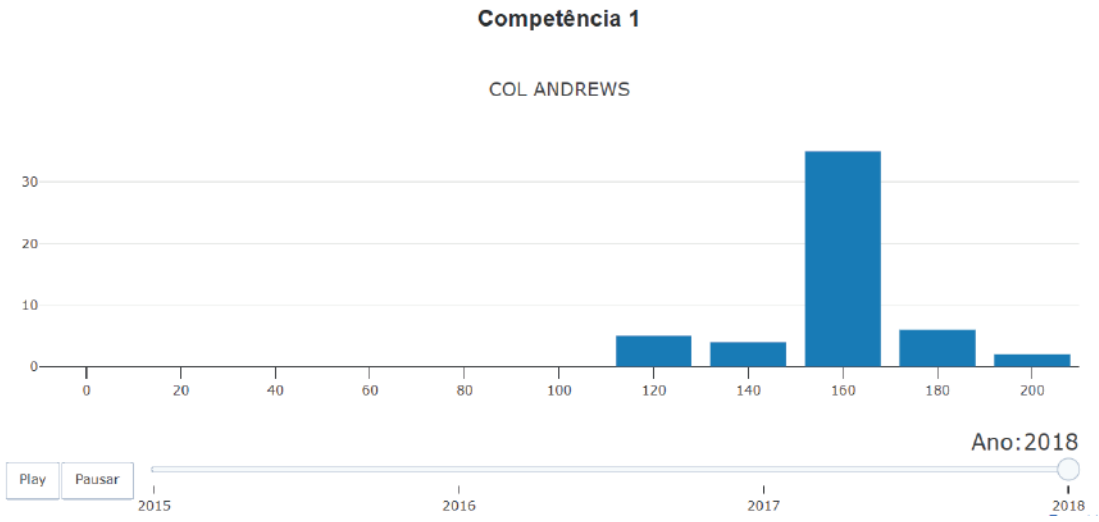


Figura 7. Distribuição das Notas em cada Competência

O gráfico de notas máxima e mínima por número de acertos (Fig. 8) apresenta as notas obtidas pelos alunos pela TRI (Teoria de Resposta ao Item) e o número de acertos.

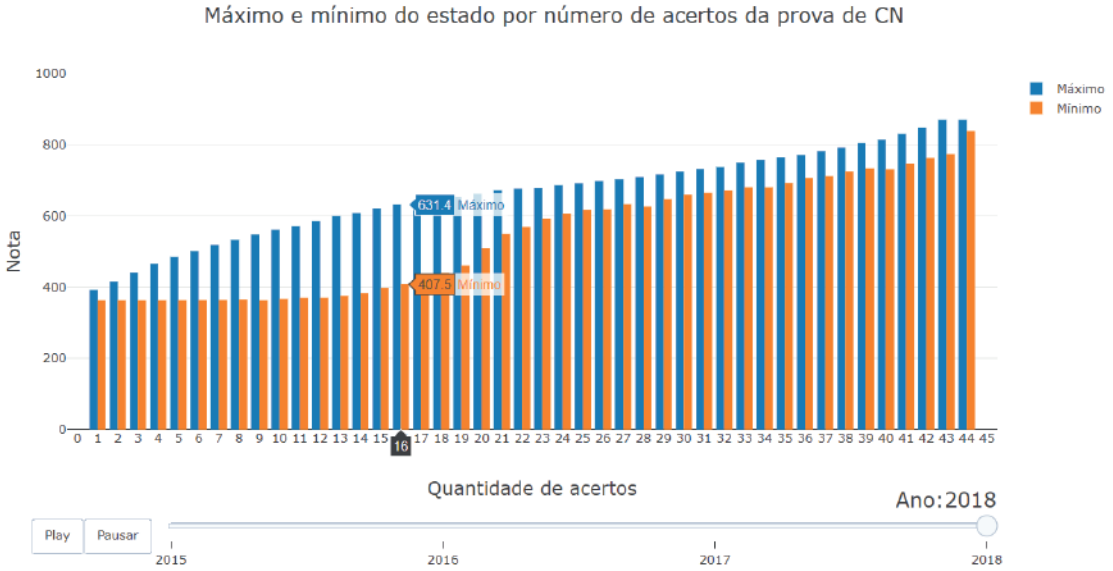


Figura 8. Notas Máximas e Mínimas por Número de Acertos

As questões são divididas em níveis fácil, médio e difícil. Pelo padrão de erros e acertos do candidato, é possível prever *chutes*. Desse jeito, a nota final não depende só do número de acertos e, sim, do nível de dificuldade das questões. Esse gráfico mostra como ficaram distribuídas as notas máxima e mínima de uma questão após a aplicação do

TRI. No exemplo em questão, aqueles que acertaram a questão 16 da prova de Ciências da Natureza e suas Tecnologias podem ter suas notas variando entre 407,5 até 631,4.

Como último exemplo, o gráfico de porcentagem de acertos por habilidade na Figura 9 apresenta no eixo horizontalas habilidades de cada área de conhecimento, numeradas de 1 a 30 e, no eixo vertical, as porcentagens de acertos pelos alunos de um colégio, da(s) questão(ões) que apresentaram a habilidade; o diâmetro de cada bolha é proporcional ao número de vezes em que a habilidade foi contemplada na prova.

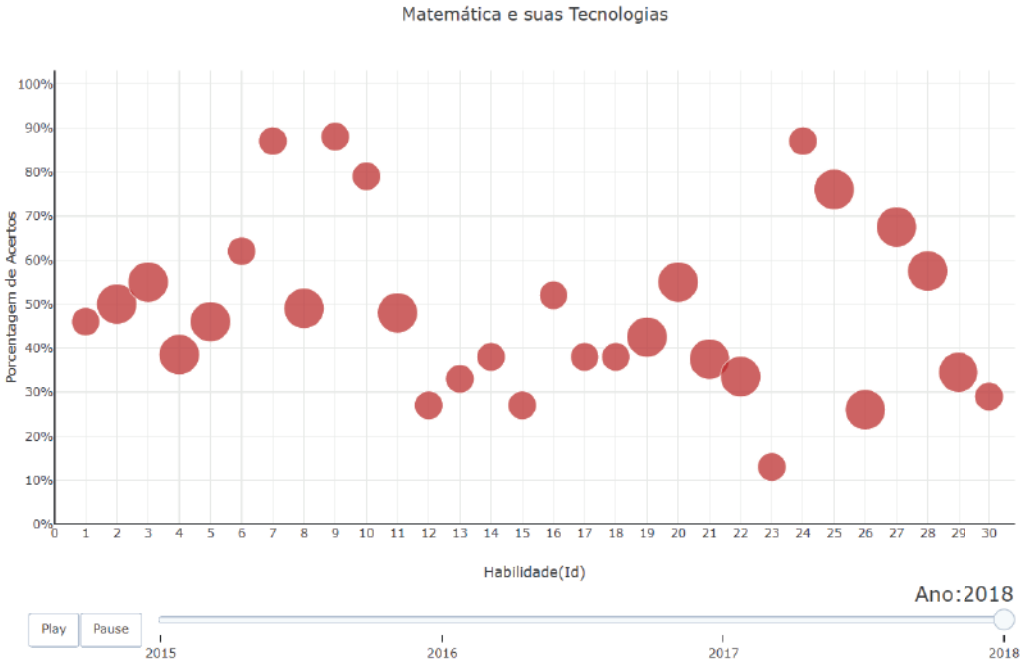


Figura 9. Porcentagem de Acertos por Habilidade

5. Comentários Finais

Apresentamos o projeto de modelagem, integração e visualização dos resultados das provas do ENEM publicados pelo INEP. Descrevemos o software desenvolvido e alguns dos desafios encontrados para a carga de dados. A aplicação está disponível em **https://enem.biobd.inf.puc-rio.br/** e vem sendo testada por usuários convidados. É possível analisar provas de anos diferentes simultaneamente, comparar duas escolas, fazer análises por estado ou do Brasil inteiro. Nossa solução permite a análise e a criação de estratégias para aprimoramento de metodologias de ensino e aprendizagem.

Referências

de Frias, J. L. D. (2015). Uma ferramenta para a obtenção e análise de dados do enem. `https://www.maxwell.vrac.puc-rio.br/25352/25352.PDF`.

de Souza, V. A. L. (2018). Sintonia fina de um banco postgresql - estudo de caso enem. *Trabalho de Conclusão de Curso - Engenharia de Computação - PUC-Rio*.

INEP (2019). Microdados enem. `http://portal.inep.gov.br/microdados`.

Society, T. I. (2019). Common format and mime type for comma-separated values (csv) files. *Available at `https://tools.ietf.org/html/rfc4180`*.

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Dataset Show Case

PROCEEDINGS

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

Program Chair

Mirella M. Moro

Editorial

It is a pleasure to introduce the proceedings of the SBBD Dataset Showcase Workshop – DSW. It is the second time such an event happens with SBBD, and we are proud to see the database community participation through papers submissions and reviewers' engagement.

The Dataset Showcase Workshop purpose is to provide a forum for sharing and discussing how to build and organize datasets that serve as basis for research work developed in the database community. The contribution of papers published at DSW is the final product in the form of a dataset, usually extracted from some database or Web platform, cleaned, transformed and processed, often enhanced with external data and able to be reused for experiments reproduction as well as amplified to other scenarios.Furthermore, the DSW provides a real possibility of improving collaboration between different research groups through sharing data used in scientific endeavours.

Regarding the evaluation process, all submitted papers were evaluated by at least three members of the DSW program committee. For its second edition, DSW received 12 submissions. Due to their high quality and interesting datasets, 10 submissions were accepted to be published in this proceedings and presented at the Workshop.

Finally, as SBBD DSW co-chairs, we would like to thank the authors and their collaborators for submitting their work to the workshop. Likewise, we really appreciate the reviewers work for the precious time spent in the careful evaluation of all submissions. We would also like to thank SBBD organizers for their outstanding support. We wish the community an excellent workshop and success in their future work. Welcome to the second edition of SBBD DSW, and we hope you enjoy the workshop.

Mirella M. Moro (Universidade Federal de Minas Gerais)
Renata Galante (Universidade Federal do Rio Grande do Sul)
DSW Program Chairs

34th Brazilian Symposium on Databases

October 7-10, 2019
Fortaleza - CE - Brazil

Promotion

Sociedade Brasileira de Computação – SBC
Comissão Especial de Banco de Dados (CEBD) da SBC

Organization

Departamento de Computação, Universidade Federal de Ceará– UFC

SBBD Steering Commitee

Ângelo Brayner (UFC)
Bernadette Lóscio (UFPE) coordenadora da CEBD
Carina Dorneles (UFSC)
Sérgio Lifschitz (PUC-Rio)
Fábio Porto (LNCC)
Carmem Hara (UFPR)

SBBD 2019 Commitee

Steering Committee Chair
Bernadette Lóscio (UFPE)

Local Chair
José Maria da Silva Monteiro Filho (UFC, Brazil)

Full Paper Chair
Carina F. Dorneles (UFSC, Brazil)

Short Paper Chair
Fábio Porto (LNCC, Brazil)

Demos and Applications Chair
Robson L. F. Cordeiro (ICMC-USP, Brazil)

Thesis and Dissertation Workshop Chair
Jonice Oliveira (UFRJ, Brazil)

Tutorials Chair
Altigran Soares da Silva (UFAM, Brazil)

Short course Chair
Maria Cláudia Cavalcanti (IME, Brazil)

Workshop Chair
José Antônio Macedo (UFC, Brazil)

Thesis and Dissertation Contest Chair
Caetano Traina Jr. (USP, Brazil)

Graduation Student Workshop Chair
Ticiana Linhares (UFC, Brazil)

Local Organization Committee
SBBD Local Chair: José Maria da Silva Monteiro Filho (DC/UFC)
Leonardo Oliveira Moreira (Instituto UFC Virtual/UFC)
Marum Simão Filho (UNI7)
Angelo Roncalli de Alencar Brayner (DC/UFC)
Javam de Castro Machado (DC/UFC)

Graduation Student Workshop Program Committee
Anderson Ferreira (UFOP)
Daniel H Dalip (CEFET MG)
Daniel de Oliveira (UFF)
Daniel Kaster (UEL)
Daniel Lichtnow (UFSM)
Eduardo Ogasawara (CEFET RJ)
Flávio Coutinho (CEFET MG)
Giseli R. Lopes (UFRJ)
Helena Grazziotin (UCS)
Isabela Gasparini (UDESC)
Jussara Almeida (UFMG)
Luciano Barbosa (UFPE)
Michele Brandão (IFMG)
Ronaldo S. Mello (UFSC)
Sérgio Lifschitz (PUC Rio)
Ticiana Coelho da Silva (UFC)
Vanessa Braganholo (UFF)

Table of Contents (Dataset Show Case)

BDSmells Data Set: A PL/SQL Code Smell Data Set

Antônio Diogo Forte Martins[1], **Tiago da Silva Vinuto**[1],
José Maria da Silva Monteiro Filho[1], **Javam de Castro Machado**[1]

[1]Department of Computing, Federal University of Ceará, Fortaleza-Ceará, Brazil

{diogo.martins, jose.monteiro, javam.machado}@lsbd.ufc.br

tiagovinuto@gmail.com

Abstract. *Code Smell can be defined as any feature in the source code of a software that may indicate possible problems. In database scenario, the term Bad Smell has been used as a generalization of Code Smell, once some features that are not directly related to code also can indicate problems, such as, inefficient queries. Bearing in mind the recurrence of different Bad Smell, they were catalogued. In this paper, we provide a public data set, called BDSmells Data Set, which contains the occurrence of PL/SQL Code Smells extracted from 20 open source projects collected on GitHub. The BDSmells Data Set can be useful to perform data analysis to discover links between smells, and to assist database professionals to avoid future problems when developing PL/SQL projects.*

1. Introduction

According to Walter et al. [2018], Code smells [Fowler 2018] indicate possible flaws in software design. This term refers to the ”smell” of a code, and may indicate possible problems related to structure, efficiency, maintenance and readability of a source code. However, even though a Code Smell does not directly represent a defect in the source code, it is not technically incorrect and does not interfere with code execution [Singh and Chopra 2013], it can not be ignored, once it may cause future problems and compromise the software quality.

A database plays a central role in the architecture of an information system. The effective use of database affects fundamental quality parameters, such as performance and maintainability, of these systems. In the database context, the term Bad Smell has been used as a generalization of the term code smell, once some characteristics that are not exclusively related to the software code may also indicate problems, for instance, inadequate type of an index structure or a SQL query written in a inefficient way. More specifically, the term SQL code smells has been used to indicate ‘smelly’ SQL commands.

Oracle Corporation is one of the largest vendors in the database market. Its flagship product is the Oracle Database, a relational database management system (RDBMS). PL/SQL (Procedural Language for SQL) is Oracle Corporation’s procedural extension for SQL and the Oracle relational database. PL/SQL includes procedural language elements such as conditions and loops. It allows declaration of constants and variables, procedures and functions, types and variables of those types, packages, arrays and triggers. It can handle exceptions (runtime errors).

Recently, different tools were developed to automatically identify SQL code smells occurrences in a given PS/SQL code. With the help of these tools, it has become

possible to perform studies about the prevalence of PL/SQL code smells using the reports generated by these tools. Despite the extensive availability of open-source PL-SQL code repositories and static code analyzers, to the best of our knowledge, there is not a public data set for this purpose.

In this paper, we provide a public data set, called BDSmells Data Set, which contains the occurrence of PL/SQL Code Smells extracted from 20 open source projects collected on GitHub. The BDSmells Data Set can be of great help to assist database professionals to avoid future problems when developing PL/SQL projects. It may also be useful for other people wishing to perform data analysis in order to discover patterns and links between smells.

2. Related Work

In Palomba et al. [2015], the authors proposed a WEB platform that allows users to share and validate data sets of code smells, which is called LANDFILL. The platform proposed by Palomba et al. [2015] is intended primarily to optimize existing data sets. An example of optimization proposed by the authors would be to add missing instances of code smells, indicating possible misclassified instances. The data set used by the authors to validate their proposal consists of 243 instances of five types of code smells which are: Divergent Change, Shotgun Surgery, Parallel Inheritance, Blob and Feature Envy. According to Palomba et al. [2015], the used data set was built from 20 open-source software projects.[1]

In Yamashita and Moonen [2013], the authors presented a study designed to provide empirical evidence on how developers perceive code smells. According to Yamashita and Moonen [2013], the developed study helps to identify the smells that are considered more relevant by software developers and, from this identification, recommend them with higher priority. In Yamashita and Moonen [2013], the authors used a data set consisting of 12 types of code smells extracted from three open source Java projects. [2]

In the work presented by Sharma et al. [2017], the authors investigated the fundamental characteristics of code smells by performing an empirical study carried out on a set of code smells, which, according to the authors, are the most frequent. In addition, the authors examine the correlation between two categories of code smells: Design Smells and Implementation Smells. Sharma et al. [2017] used a data set consisting of 19 design smells and 11 implementation smells in 1988 open source C# projects. All the projects combined are composed of more than 49 million lines of code. [3]

Additionally, there are other works that also study code smells. Singh and Chopra [2013] presented a study related to the code smells and discussed methods to detect them, as well as, analyze tools that ease the smells automatic detection. Khumnin and Senivongse [2017] proposed a tool that aims to automate the detection of design antipatterns of logical databases. These works deserve to be remembered to demonstrate the relevance of the code smells subject, although they do not focus on the construction of data sets.

Our proposed data set is different from the previous mentioned because these data

[1]The platform is public and available at www.sesa.unisa.it/landfill/.

[2]The data set is public and available at https://data.mendeley.com/datasets/8n6k8dfw2f/1

[3]The data set is public and available at https://zenodo.org/record/2538646#.XRtiZy5KikB

sets contain object oriented Code Smells. Our focus is to provide a data set containing Code Smells present in PL/SQL code and there are not any other public data set of PL/SQL code smells available as far as we are concerned.

3. Data Set Contextualization

In database context, the term Bad Smell has been used as a generalization of the term code smell, once some characteristics that are not exclusively related to the software code may also indicate problems, for instance, inadequate type of an index structure or a SQL query written in a inefficient way. Karwin [2010] presents a SQL antipattern catalog, that is, a collection of common issues developers frequently encounter while working with SQL. Besides, he categorizes SQL antipatterns into four categories: Logical Database Design, Physical Database Design, SQL Query, and Application Development. The SQL Query antipatterns are called SQL code smells since they are specific to SQL queries and indicate 'smelly' code, i.e. there might be a bug or an issue.

In Sharma et al. [2018] the authors define database smells as "the characteristics of database code (either DDL, DQL or DML SQL statements), database system, or stored data that indicate violation of the recommended best practices and potentially affect the quality of the software system in a negative way". They categorize database smells in three categories: Schema smells, Query smells and Data smells. Schema smells arise due to poor schema design. Compound attribute, index abuse, and god table are examples of database schema smells. Query smells arise from poorly written SQL queries. Misused null (when null is used as an ordinary value in SQL queries) and non-grouped column reference (when a query references at least one non-grouped column in the presence of group by clause) are examples of query smells. Data smells arise from poor data handling in databases. Intermingled data types (where numbers and alphabets are intermingled leading to confusion and subtle bugs; for instance, using 'O' instead of '0' in 7O34) is an example of data smells.

4. Application

In de Almeida Filho et al. [2019], the authors used the BDSmells Data Set in an exploratory and empirical study about the prevalence of PL/SQL code smells in PL/SQL open-source projects. The authors concluded that:

- Some PL/SQL Code Smells occur more than others, while some do not occur in any of the selected open-source projects.
- Many PL/SQL Code Smells are strongly correlated, reaching a correlation coefficient of over 0.9 in some cases.
- The authors explored the Apriori algorithm and found exciting association rules indicating that PL/SQL Code Smells occur together.
- The PL/SQL Code Smells on the data set are organized in two main clusters.

In de Almeida Filho et al. [2019], the authors state that the results of the study have the potential to aid database professionals to avoid future problems while developing PL/SQL projects, because the results show the predominant PL/SQL Code Smells among the analyzed projects and the association rules that they found using the BDSmells Data Set.

5. Data Set Design

Our aim is to provide a data set of PL/SQL code smells collected from open-source projects. In order to perform this task, we followed the workflow presented in Figure 1.

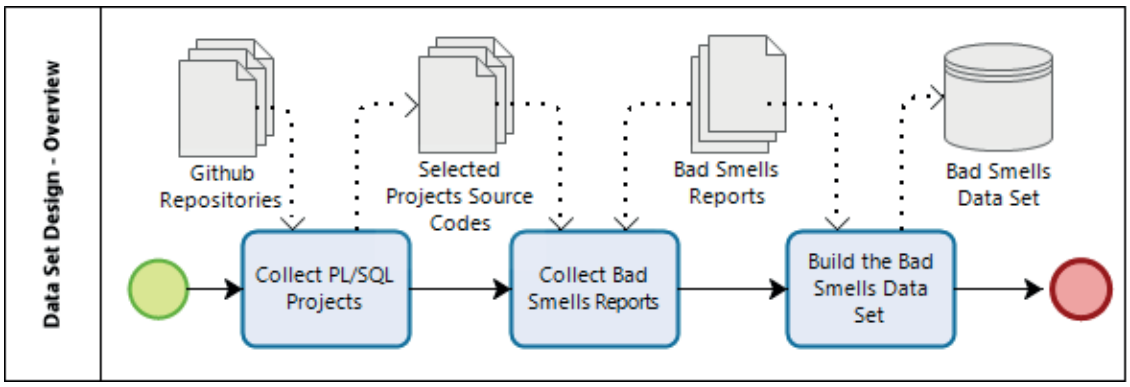


Figure 1. Data Set Design - Overview

5.1. Collect PL/SQL Projects

We searched for large scale PL/SQL projects on GitHub to collect the SQL code smells. The chosen projects repositories were selected based on 3 criteria: the number of GitHub stars, which represents the amount of developers who appreciate the repository; the number of repository forks, which represents the amount of times that the repository was copied for developers private repositories; and the number of commits, which actually are real code enhancement like new features or code refactoring, on the projects repository. The selected projects alongside with their number of stars, number of forks, lines of code and number of developers are listed in Table 1.

5.2. Collect Bad Smells Reports

In this step, we used the Manduka Tool [4] to analyzed the PL/SQL code and collect the SQL Code Smells. The Manduka Tool performs static code analysis and detect the SQL code smells based on several rules already defined in the software. The Manduka rules are classified in five categories: readability, maintenance, code correction, code structure and efficiency. These rules are labeled from 1 to 50, and their descriptions are available alongside the BDSmells Data Set.

5.3. Build the Bad Smells Data Set

Each Manduka Tool analysis generates a XML file containing all the information about the SQL code smells found in an specific project. These reports contain the local where each smell can be found in the source code, the ID, description and severity of the SQL code smell. In possession of all the reports, we developed a Python script to read the reports and extract the number of occurrences of each smell in the projects in order to build the data set. In Figure 2, we have a sample of a XML report file that is generated when the Maduka Tool finishes an analysis.

[4]https://github.com/VirtusaPolarisGTO/manduka

Table 1. Selected Projects

Project	Stars	Forks	Lines of Code	# of Devs.
alexandria-plsql-utils	293	132	1878	16
OpenML	183	34	300474	19
Logger	144	77	3406	8
generate-sql-merge	95	49	472	11
oos-utils	77	33	8014	6
sqld360	43	18	22692	1
PLSQL-JSON	39	10	2374	1
ssis-queries	36	20	443	2
sha256_plsql	22	11	553	1
tePLSQL	22	6	1077	2
tapiGen2	15	9	1077	1
plsql-aws-s3	13	2	129	1
apex-plugin-apextooltip	9	2	5423	1
apex-plugin-templates	9	0	672	1
mailgun-plsql-api	8	4	979	1
dbax-lite	8	2	1204	1
apex-plugin-apexscreencapture	7	2	13010	1
jwt_ninja	6	0	302	2
plsqlstarter	5	2	65380	1
method5	5	0	5184	1

```
<violations>
<file name="work\ast\ora\xml_stylesheet_pkg.pks">
<violation>
<message>Restrict use of DEFAULT clause in parameter declarations</message>
<line>25</line>
<column>46</column>
<type>Tech-Defects</type>
<severity>2</severity>
<rule id="23" name="Restrict use of DEFAULT clause in parameter declarations"/>
</violation>
<violation>
```

Figure 2. A sample from a XML report generated by the Manduka Tool.

All the generated XML files were parsed using the Python module *xml.etree.ElementTree* and then the number of occurrences could be easily computed. With the number of occurrences of each smell in each project, a Pandas *DataFrame* [McKinney 2010] was created to store the obtained values. During this step, we needed to pay attention to the missing values, because neither all smells occur in all projects.

6. Data Set Description

The BDSmells Data Set is provided in a *csv* file where the columns are the projects names and the rows are the ID of a SQL code smell. The scope of the code smells analysis is the project as a whole, that is, each row of the data set has the sum of the correspondent code smell across all files.

In Figure 3, we have a sample of the data set to better illustrate how the data is organized. Using the *alexandria-plsql-utils* project as an example, by analyzing the

smell_id/Project	alexandria-plsql-utils	apex-plugin-apexscreencapture	apex-plugin-apextooltip	apex-plugin-templates	db
1	441	0	0	6	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	
5	2	0	0	0	
6	5	0	0	0	
7	9	0	0	0	
8	0	0	0	0	
9	0	0	0	0	
10	22	2	2	6	
11	0	0	0	0	

Figure 3. A sample from the BDSmells Data Set.

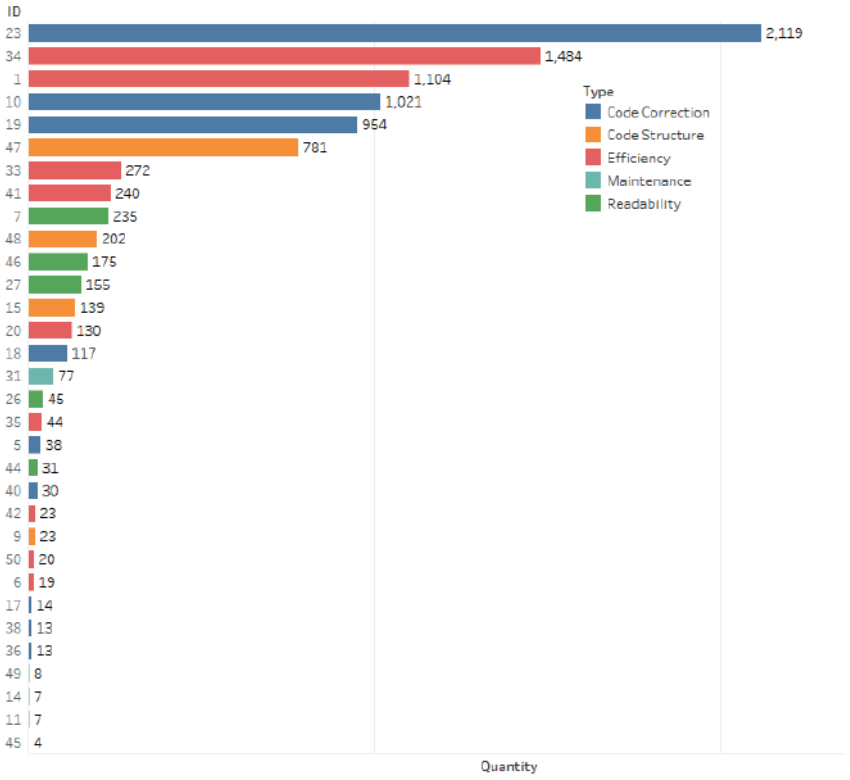


Figure 4. The quantity of SQL code smells across all projects.

figure, we can note that it has 441 occurrences of the PL/SQL Code Smell ID 1 and 0 occurrences of the PL/SQL Code Smell ID 9.

In Figure 4, we can observe that the SQL code smell ID 23 is the most occurring across all projects. This SQL code smell refer to the use o the *DEFAULT* clause when declaring a parameter, but in order to avoid problems with *NULL* values, the developers tend to use it and violate this recommendation. It is also important to highlight that Code Correction and Efficiency smells are the most occurring types of smells in the selected projects.

The SQL code smells of identifiers (ID) 2, 3, 4, 8, 12, 13, 16, 21, 22, 24, 25, 28, 29, 30, 32, 37, 39 and 43 are not present in Figure 4 because they were not found in the analyzed projects. There are many explanations about how this happened. For instance,

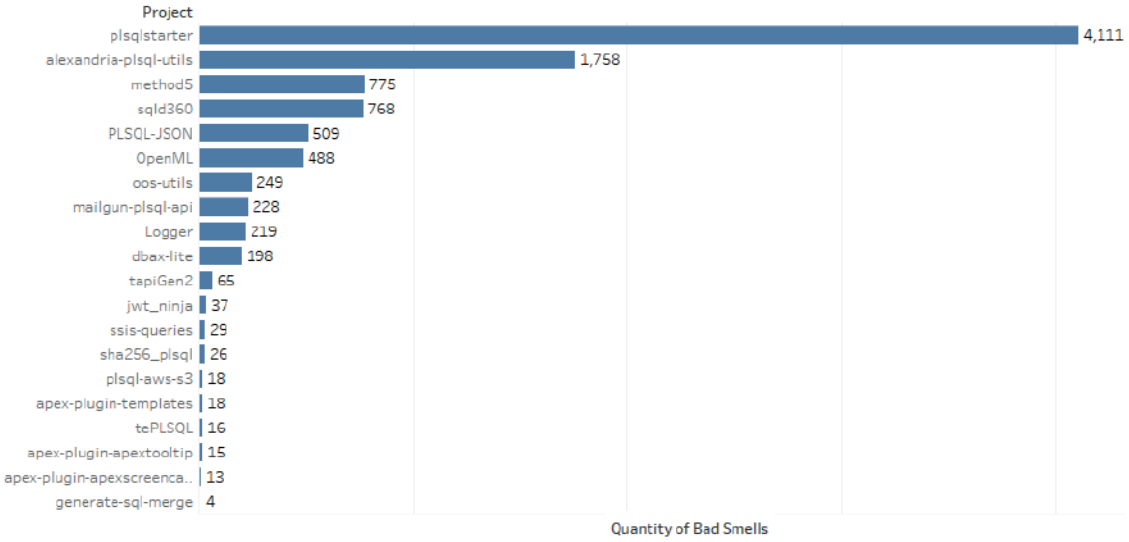


Figure 5. The quantity of SQL code smells for each project.

the SQL code smells 2, 8, 13, 16 and 21, are rare since they use specific Oracle packages. There are also some smell, like SQL code smell 3, that is already avoided by the main IDEs. Other SQL code smells rules, such as 12, 22, 24 and 30 are already well known in the community, so the programmers easily avoid them.

Figure 5 presents the quantity of SQL code smells that occur in each project. We can observe that the projects *plsqlstarter* and *alexandria-plsql-utils* have combined 61.5% of all the SQL code smells occurrences. Note that *alexandria-plsql-utils* has 132 forks and 16 contributors. So, it is a complex project. On the other hand, the *plsqlstarter* project has 65,380 lines of code. Thus, it is a very large project.

Using the project *plsqlstarter* as an example, the distribution of PL/SQL Code Smells for *plsqlstarter* project can be seen in Figure 6. In the *plsqlstarter* project, 26 out of the 50 SQL code smells occur, and there are six PL/SQL Code Smells with a high number of occurrences which the IDs are: 1, 10, 19, 23, 34, 47. These PL/SQL Code Smells with high number of occurrences in the *plsqlstarter* project are related to Code Correction, Efficiency and Code Structure. We can affirm analyzing Figure 6 that *plsqlstarter* project is, alone, responsible for 80.40% and 48.92% of the occurrences of the PL/SQL Code Smells ID 19 and ID 34, respectively.

7. Conclusion

In this work, we presented a public data set, called BDSmells Data Set, which contains the PL/SQL Code Smells present in 20 public open source project from GitHub repositories. We discussed all the steps of the proposed workflow to build the data set properly. Also, we provided an overview of the data set, we highlighted the PL/SQL Code Smells that did not occur in any project and the most occurring collected code smells along all projects. We believe that this data set may help developers to avoid future problems when developing PL/SQL software project. In de Almeida Filho et al. [2019], the authors used the BDSmells Data Set in an exploratory and empirical study about the prevalence of PL/SQL code smells in PL/SQL open-source projects. As for future works, we intend to

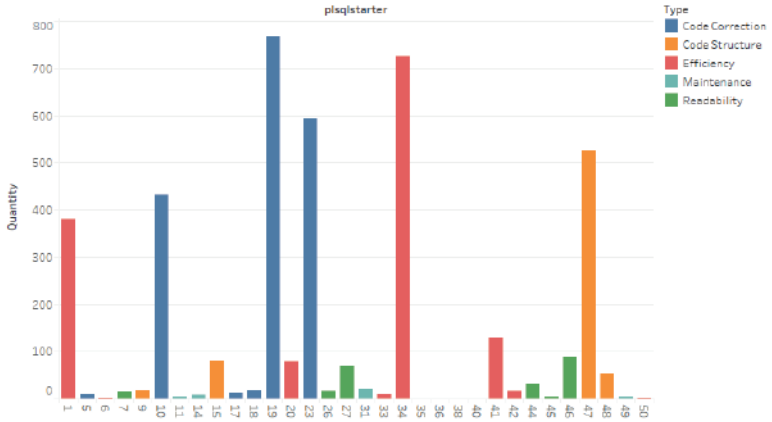


Figure 6. The distribution of SQL code smells in the *plsqlstarter* project.

build a new data set treating each file of a project individually increasing the granularity and consequently increase the accuracy of the results.

The BDSmells Data Set is public for research, available at `https://bit.ly/2KNPGv5` (DOI: 10.5281/zenodo.3258678). In case this data set is used for scientific, or academic purposes, or in case it is publicly mentioned for whatever purpose, please include the citation to this work.

Acknowledgements

This research was funded by LSBD/UFC.

References

de Almeida Filho, F. G., Martins, A. D. F., Vinuto, T. d. S., Monteiro, J. M., de Sousa, I. P., de Castro Machado, J., and Rocha, L. S. (2019). Prevalence of bad smells in pl/sql projects. In *Proceedings of the 27th International Conference on Program Comprehension*, ICPC '19, pages 116–121, Piscataway, NJ, USA. IEEE Press.

Fowler, M. (2018). *Refactoring: improving the design of existing code*. Addison-Wesley Professional.

Karwin, B. (2010). *SQL Antipatterns: Avoiding the Pitfalls of Database Programming*. Pragmatic Bookshelf, 1st edition.

Khumnin, P. and Senivongse, T. (2017). Sql antipatterns detection and database refactoring process. In *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 199–205.

Lanza, M. and Marinescu, R. (2007). *Object-oriented metrics in practice: using software metrics to characterize, evaluate, and improve the design of object-oriented systems*. Springer Science & Business Media.

McKinney, W. (2010). Data structures for statistical computing in python. In van der Walt, S. and Millman, J., editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56.

Mens, T. and Tourwé, T. (2004). A survey of software refactoring. *IEEE Transactions on software engineering*, 30(2):126–139.

Moha, N., Gueheneuc, Y.-G., Duchien, L., and Le Meur, A.-F. (2009). Decor: A method for the specification and detection of code and design smells. *IEEE Transactions on Software Engineering*, 36(1):20–36.

Palomba, F., Bavota, G., Penta, M. D., Oliveto, R., and Lucia, A. D. (2014). Do they really smell bad? a study on developers' perception of bad code smells. In *2014 IEEE International Conference on Software Maintenance and Evolution*, pages 101–110.

Palomba, F., Di Nucci, D., Tufano, M., Bavota, G., Oliveto, R., Poshyvanyk, D., and De Lucia, A. (2015). Landfill: An open dataset of code smells with public evaluation. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 482–485.

Sharma, T., Fragkoulis, M., Rizou, S., Bruntink, M., and Spinellis, D. (2018). Smelly relations: Measuring and understanding database schema quality. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, ICSE-SEIP '18, pages 55–64, New York, NY, USA. ACM.

Sharma, T., Fragkoulis, M., and Spinellis, D. (2017). House of cards: Code smells in open-source c repositories. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 424–429.

Singh, G. and Chopra, V. (2013). A study of bad smells in code. *Int J Sci Emerg Technol Latest Trends*, 7(91):16–20.

Umesh, I. and Srinivasan, G. (2015). A study on bad code smell.

Walter, B., Fontana, F. A., and Ferme, V. (2018). Code smells and their collocations: A large-scale experiment on open-source systems. *Journal of Systems and Software*, 144:1–21.

Yamashita, A. and Moonen, L. (2013). Do developers care about code smells? an exploratory survey. In *2013 20th Working Conference on Reverse Engineering (WCRE)*, pages 242–251.

Beyond Tears and Smiles with *ReactSet*: Records of Users' Emotions in Facebook Posts

Mirela T. Cazzolato[1], Felipe T. Giuntini[1], Larissa P. Ruiz[2], Luziane de F. Kirchner[3], Denise A. Passarelli[2], Maria de Jesus Dutra dos Reis[2], Caetano Traina-Jr.[1], Jó Ueyama[1], Agma J. M. Traina[1]

[1] Institute of Mathematics and Computer Sciences
University of São Paulo - São Carlos, SP, Brazil,
[2]Department of Psychology
Federal University of São Carlos - São Carlos, SP, Brazil,
[3]Dom Bosco Catholic University, Campo Grande, MS, Brazil
{mirelac, felipegiuntini}@usp.br,
{caetano, joueyama, agma}@icmc.usp.br

Abstract. *Emotion and feelings recognition have been studied in a wide research range in the past decades. The advancement and spread of social networks and online applications, such as Facebook, Instagram, and Whatsapp has motivated the sharing of news and communications among users. Specifically, in social networks, users can give reactions related to different contents, which can provide meaningful clues to study emotions. In this work, we present ReactSet, a dataset composed of records of users' emotions for different news, along with some de-identified personal information and the time they took to analyze each news. All the news were selected by specialists from the areas of Psychology and Human-Behavior analysis. We describe ReactSet contents, present statistics, potential applications, and challenges regarding the collected data. ReactSet is publicly available for research use, under the Creative Commons license.*

1. Introduction

Social relations and networking are fundamental components of human life, historically linked according to limitations from time and space. However, the technological evolution, the Internet, and its diffusion of use have partially removed such restrictions. The emergence of Web technologies allowed services in virtual social networks as they exist in a non-virtual way. This notion of social networking and its methods of analysis has attracted significant interest and curiosity from the community at large in the last decades, especially from the areas of social and behavioral sciences. The interest derives from social networks providing a powerful abstraction of the structure and dynamics of different types of people or person-technology interaction. Social network analysis can indicate the study of virtual social structures and its effects to analyze social and cultural aspects.

People have increasingly prioritized agile, fast, and efficient communication. The form of peer communication in social networks and online applications such as Facebook, Instagram, Whatsapp, and Twitter has also been transformed and reduced each day to use fewer characters. In this virtual context, new notations formed by combining a set of alphanumeric characters and punctuation, such as :) , (+_+) , O_o , :(, indicate objects, gestures and especially expressions of opinions and emotions. These annotations

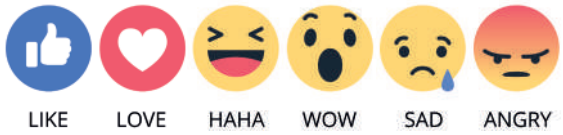


Figure 1. Six Facebook reactions, currently available at Facebook.

have been evolving and enhanced with graphical features such as colors (emojis) and animations (emoticons). Emoticons are small images or combinations of diacritic symbols that participants of social media can use to express feelings like excitement, anxiety, and anger. The content of posts, messages, and comments can include emoticons. Recently, the social network Facebook selected and restyled a set of more significant emoticons, and named them reactions. The reactions make it possible for users to react and express themselves emotionally to the published postings. Figure 1 shows the Facebook reactions Like, Love, Haha, Wow, Sad, and Angry.

Emotional facial expressions are powerful stimuli in the analysis of social interactions. Individuals use those expressions to identify common hazards, avoid conflicts, keep track of partners' emotional reactions, adjust their behavior depending on others' attitudes, learn to describe emotions verbally, and so on [Ekman et al. 1969, Ekman and Friesen 1971, Lerner and Keltner 2001, Russell et al. 2003, Spoor and Kelly 2004, Waller et al. 2008]. Although some theorists argue that such expressions are distinctly shaped at each culture [Fridlund 1994], others have pointed to their universal recognition [Ekman 1993, Frijda et al. 2000]. Charles Darwin was the first scientist to systematically evaluate the universality of facial expressions to recognize emotions, when in 1872 he wrote "The Expression of the Emotions in Man and Animals". Darwin's methods of study were based on the observation of the facial expressions of young children, as they interact with their peers, blind or sighted children, and people with mental disorders. From some experiments, Darwin found evidence showing that facial expressions correspond to specific emotions, regardless of the learning environment of each individual. In 1960, Paul Ekman continued the studies of this subject, which led him to conclude that there are a limited number of facial expressions that signaled emotions. In 1982 he postulated about the universality of expression of six basic emotions (anger, disgust, fear, joy, sadness, and surprise), and in 1990 he argued about the existence of eleven other universal emotions (fun, contempt, contentment, embarrassment, excitement, guilt, pride in conquest, relief, satisfaction, sensory pleasure, and shame). However, the latter still presents concerns regarding its universality [Wolf 2015].

The research on emotional expressions can be applied to different kinds of communication, and some of them are usually the most significant social media interactions. People are continually using social networks to express what they feel [Jibril and Abdullah 2013]. Facebook has about 2,38 billion monthly active participants, who spend most of their day online, making the virtual environment a useful source of evidence about what participants consider and feel about [Statista 2019, Vashisht and Thakur 2014]. Emoticons are currently the most broadly used media in the virtual environment [Oleszkiewicz et al. 2017], addressed by several scientific literature [Wegrzyn-Wolska et al. 2016, Wang and Castanon 2015, Oleszkiewicz et al. 2017, Huang et al. 2008]. Instead of describing emotions through words, emoticons can help to

provide hints for communication that may be in the virtual technology world, i.e., non-verbal clues [Vashisht and Thakur 2014]. The importance of this declaration has given rise to the concern of scientists, regarding how emoticons can be used in several fields, such as mental health issues, reactions to stressful occurrences, preferences or decisions, and multiple opinion polls [Luor et al. 2010, Gaspar et al. 2016].

Previous use of the data. Recently, several works on the data science field have explored how well emoticons or reactions can represent expressions of corresponding emotions in real environments. In this context, emoticons and reactions are non-verbal forms of expression in online social networks. In the work [Giuntini et al. 2019], the authors employed a questionnaire on Facebook, aimed at collecting different information of users regarding specific posts. A panel of specialists in Psychology and researchers in human behavior analysis selected a set of Facebook news, with a high number of reactions. These Facebook news were made available in a survey for Facebook users. Users responded to what reaction they attributed to the posting, the polarity of feeling (negative, neutral, or positive) and could choose up to two of Ekman's most significant basic emotions (anger, disgust, fear, joy, sadness, and surprise) [Ekman et al. 1969].

Giuntini *et al.* used part of the collected users' responses to news to address how people reacted to various relevant news, at a given moment, using Facebook reactions. In this work, we present an extended and complete version of the data reported in [Giuntini et al. 2019], called *ReactSet*. *ReactSet* is a new dataset with emotions and reactions of Facebook users regarding 34 news. It provides support for different analysis involving emotions. Moreover, the dataset can serve for data mining researches that seek to solve problems related to the study of feelings, temporal patterns analysis, subjectivity, and the treatment of missing and incomplete data.

Paper outline. The remainder of this paper is organized as follows. Section 2 describes the *ReactSet* dataset. Section 3 discusses the main applications and challenges we envision to use *ReactSet*. Section 4 details how *ReactSet* is publicly available for download, along with the description of its data files in the public repository GitHub[1]. Finally, Section 5 gives the conclusion of this work.

2. *ReactSet*: Reactions of Facebook Users

In this section, we describe *ReactSet*, a dataset of Facebook users' reactions to different news.

2.1. Data Collection and Preprocessing

ReactSet was collected by the application of a questionnaire, which was publicly open to answers on Facebook during the month of July 2017. In total, 409 participants answered the questions, 149 of them with complete responses, and 260 of them with partial answers. The employed questionnaire has two parts. The first consists of personal information, such as age, sex, and scholarity. The second part asks the user to analyze different news. A total of 36 news were presented to the users, each with an image and a description. For each news, users were asked to answer the following questions:

- *Question 1.* Which reaction would you give to this post?

[1] *ReactSet* is publicly available for download at *https://github.com/mtcazzolato/reactset*.

  - *Answers*: Like, Love, Haha, Wow, Sad, Angry, using icons from Figure 1.
- *Question 2.* How do you classify this post?
  - *Answers*: Positive, Neutral, Negative.
- *Question 3.* Which is the most predominant emotion is this post?
  - *Answers*: sadness, joy, do not recognize, fear, disgust, anger, surprise.
- *Question 4.* Which is the second most predominant emotion in this post?
  - *Answers*: sadness, joy, do not recognize, fear, disgust, anger, surprise.

A group of 7 judges selected the news we used for the analysis. This group was composed of Master, Ph.D. students and specialists on human-behavior analysis. They first classified each news as containing one of the emotions: anger, disgust, fear, joy, sadness, and surprise. From the 36 news, judges selected the 24 most representative news (4 for each emotion), which presented the following characteristics [Giuntini et al. 2019]: (i) the majority of the judges found the same emotion in a single news; (ii) the majority of the participants found an emotion in the same news item. We summarize the judges' concordance for the 24 selected news in Table 1, along with the participants' concordance, separated by the observed emotion.

2.2. Data Description

Figure 2 presents the dataset schema, and Table 2 presents the tables and columns of *ReactSet*, along with their corresponding description. Three tables organize all available information:

- ***PersonalInfo.*** The users' personal information, and the dates in which the user answered the questions;
- ***News.*** The information regarding each news evaluated;
- ***AnswerNews.*** The answers of users regarding each news present in table News.

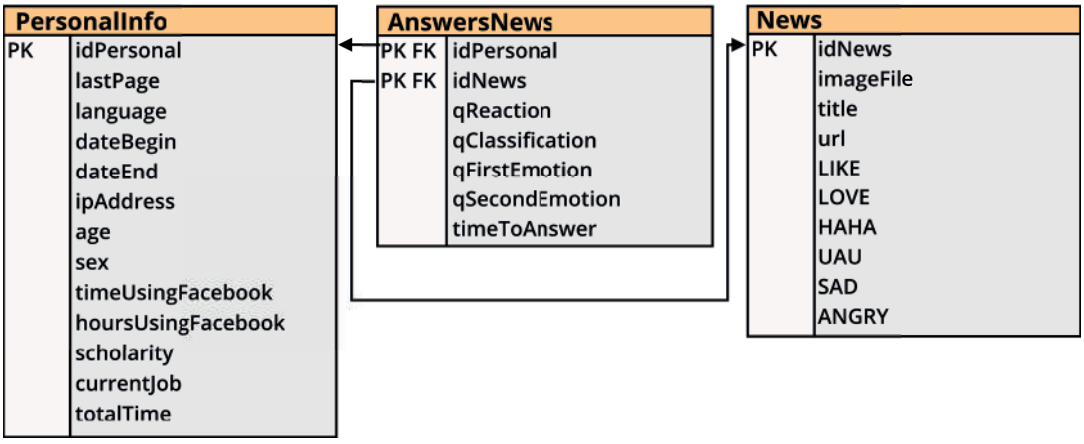


Figure 2. Schema of the *ReactSet* dataset.

In table *News*, we make available the number of reactions given by Facebook users for each available news, gathered in July, 2019. Notice that these reactions are those obtained by the Facebook post, given by the general public. The information is provided for 21 of 36 news. Figure 3 (a, b and c) shows three examples of news. Figure 3 (d) shows the collected Facebook reactions of the three news. We observe that the first news, a potential

Emotion	News ID	Participants' concordance (%)	Judges' concordance (%)
Joy	4	55.10	71.42
	10	97.96	85.71
	21	95.24	71.42
	24	90.48	85.71
Sadness	12	74.83	100
	13	85.71	85.70
	26	84.35	85.71
	27	78.23	71.42
Anger	2	63.27	71.42
	7	80.95	57.14
	23	44.22	57.14
	28	59.89	85.71
Disgust	16	23.80	28.57
	20	57.82	42.85
	30	78.91	28.58
	35	14.29	42.85
Fear	15	17.00	71.42
	29	14.97	71.42
	31	20.41	28.58
	32	7.48	57.14
Surprise	5	51.02	100
	11	70.09	57.14
	33	73.47	71.42
	25	69.39	85.71

Table 1. Users' and Judges' concordance of 24 selected news, 4 of each emotion.

terrorist attack, has received more negative reactions than the two other news, regarding sports and entertainment matters. Accordingly, we observed an agreement regarding the news content and the observed reactions.

Figure 4 (a) depicts the sex per age of the participants. From a total of 409 participants, 239 identified themselves as being a woman, 124 identified themselves as a man, 4 preferred not to inform their sex, and 42 of them did not provide an answer. We observe that women have higher age values, in comparison to men and other answers. Also, Figure 4 (b) shows that the majority of the participants are undergraduate students, followed by high school and master students. Figure 4 (c) shows the sum of reactions given by the participants, for each of the analyzed news. As we observe, the number of missing responses to the questions increased with the number of news presented by the participants. The reaction, polarity classification, and emotions of each participant according to the post can be studied in a range of applications, as we discuss next.

Table 2. Tables and columns of *ReactSet*.

Table	Column	Type	Description
PersonalInfo	idPersonal (PK)	int	Identifier of the person / answer
	lastPage	int	Last page of the questionnaire visited by the person
	language	string	Language of the questionnaire
	dateBegin	datetime	Date of when the person started answering the questionnaire
	dateEnd	datetime	Date of when the person ended answering the questionnaire
	ipAddress	string	IP address of the computer / device used to answer the questionnaire
	age	int	Age of the person
	sex	string	Sex of the person
	timeUsingFacebook	string	How long the person uses Facebook
	hoursUsingFacebook	string	Number of hours the person uses Facebook (daily average)
	scholarity	string	Complete education level
	currentJob	string	Current job situation
	totalTime	float	Total time the person spent to answer all questions
News	idNews (PK)	int	Identifier of the news
	imageFile	string	Image file name
	title	string	Title of the news depicted in the image
	url	string	URL of the news depicted in the image
	like	string	Like reactions of the news on Facebook
	love	string	Love reactions of the news on Facebook
	haha	string	Haha reactions of the news on Facebook
	uau	string	Uau reactions of the news on Facebook
	sad	string	Sad reactions of the news on Facebook
	angry	string	Angry reactions of the news on Facebook
AnswersNews	idPersonal (PK)(FK)	int	Identifier of the person / answer
	idNews (PK)(FK)	int	Identifier of the news
	qReaction	string	Reaction the person assigned to the news
	qClassification	string	Polarity classification the person assigned to the news
	qFirstEmotion	string	The most predominant emotion the person identified in the news
	qSecondEmotion	string	The second most predominant emotion the person identified in the news
	timeToAnswer	float	Time the person spent to answer the questions related to the news

(a) News ID 06. (b) News ID 25. (c) News ID 32.

(d) Number of reactions of the news on Facebook.

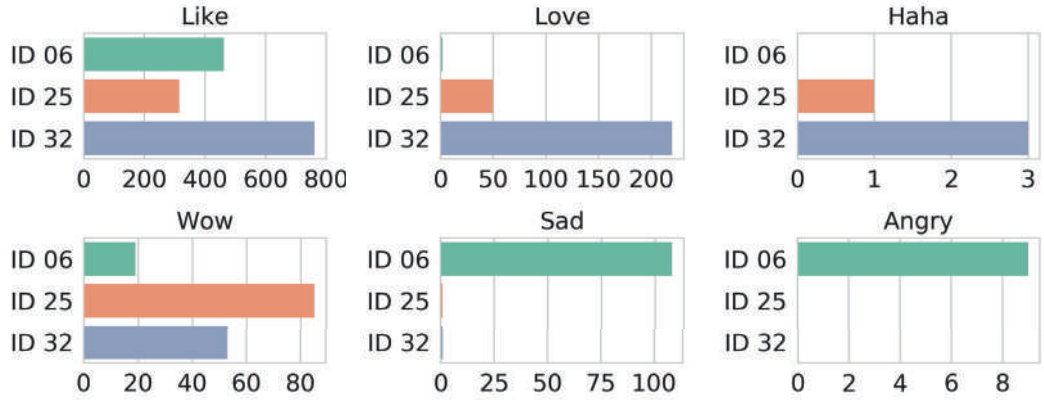


Figure 3. Example of three news present in *ReactSet*, and the number of reactions given by Facebook users.

3. Applications and Challenges regarding *ReactSet*

ReactSet opens research opportunities in various application contexts in the field of data science and analysis of feelings. We discuss potential opportunities following.

i. Temporality: For each question group, which corresponds to the questions regarding single news, *ReactSet* contains the information of how long the user took to assign the corresponding reaction, polarity classification, and emotions. This time information can allow the analysis of aspects such as excitement, motivation, and spontaneity of the participants when answering the specific news. For example, check if the user acted quickly or had to reflect on the image and title of the news.

ii. Missing data: Several researches have tried to develop new approaches so that algorithms of mining and machine learning can operate with good performance in databases with missing data. From 409 responses obtained in the survey, only 149 of the participants completed the survey. This opens up challenges, such as: (i) predicting the values of empty cells based on the user's previous responses; (ii) develop new algorithms or adaptation of existing ones to better classify feelings and emotions; (iii) address newly information, such as image content or textual news information to predict the reactions.

iii. Predictions: Considering that reactions have been used both to show affection, as well as opinion on brands and products, the prediction of reactions in social networks can help the development of more attractive, audience-guided, and positive impact ads.

iv. Small Data: For some problems, including emotions analysis, gaining knowledge

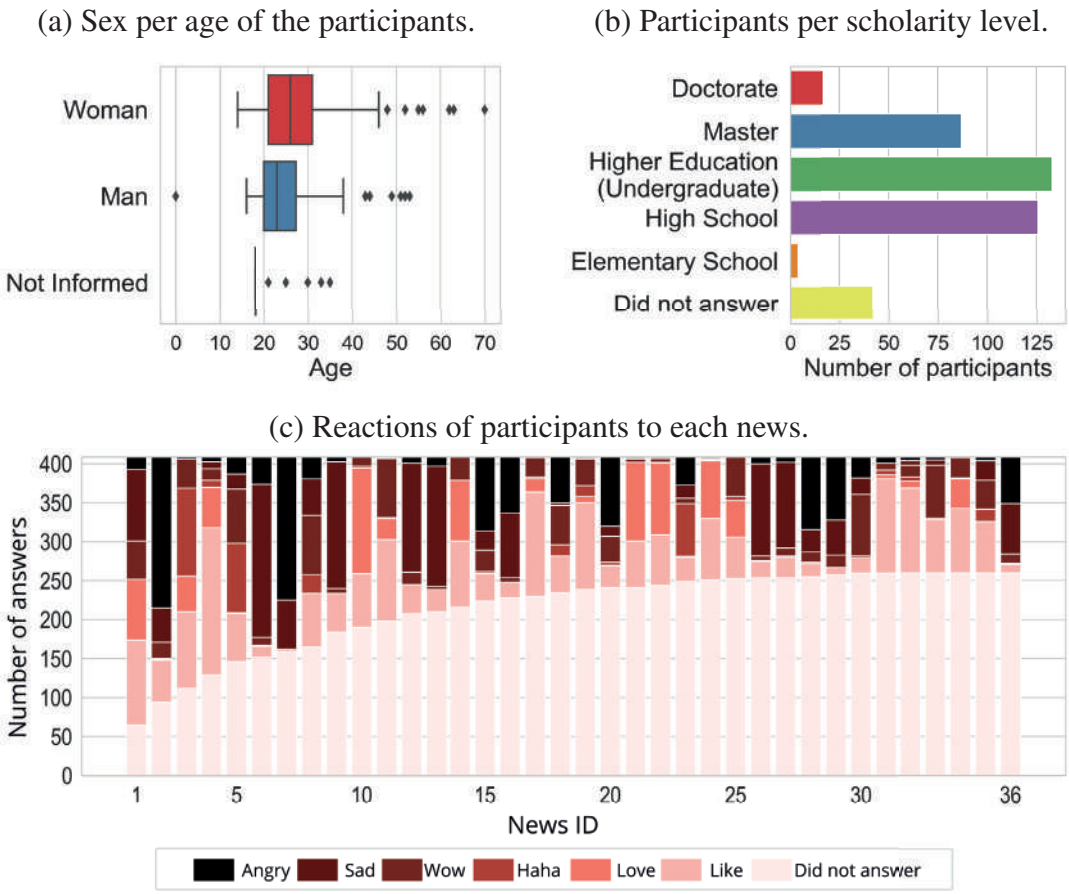


Figure 4. Overview of the participants' information and reactions to the news.

or making decisions based solely on the use of reactions can be challenging. The reactions or emoticons are just a digraph representation, which can involve motivational, excitement and even cultural aspects. For example, in the case of identifying expression of the basic emotions, as explored in [Giuntini et al. 2019] using the set of complete answers of *ReactSet*, it does not seem to be a complex task. However, from these basic emotions, other more sophisticated ones can be extracted. For this, a research effort is required, so that algorithms seek to use context information, or even performing the intersection of basic emotions. A theoretical example is provided by Figure 5 (extracted from [DeMiglio and Williams 2016]), which shows the componential model of emotions defined by [Scherer 1987], along with wellness behavior. Notice that, at the extreme of the axes (between the leaves depicted in the figure), we can see a combination of a set of emotions to define a more sophisticated emotional expression. In addition, data science researches have sought to achieve the same or better performance of techniques commonly applied in Big Data over small datasets [Faraway and Augustin 2018].

4. Download and Citation Request

ReactSet is publicly available for research use, under the Creative Commons license. The dataset is available at *https://github.com/mtcazzolato/reactset*, with the following organization of files:

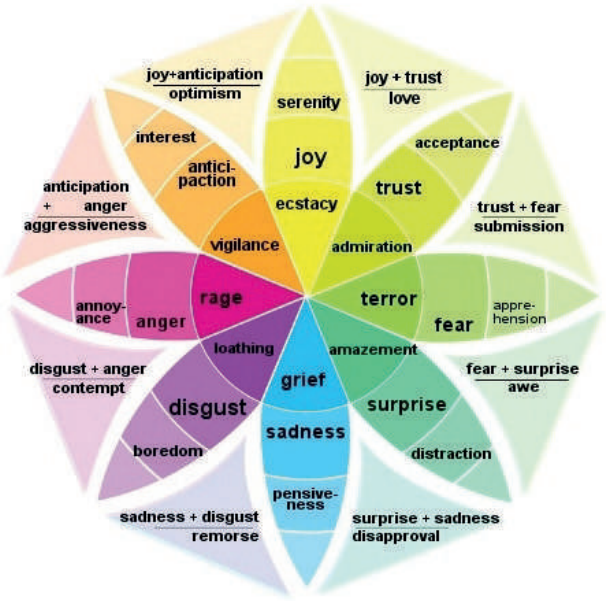


Figure 5. Compositional frame of well-being, based on the componential theory of the emotions of Scherer, extracted from [DeMiglio and Williams 2016]

- *ReactSet/*
  - *newsImages/*: folder with images of extension ".png", each corresponding to a news from table "News" (36 images).
  - *PersonalInfo.csv*: file with data from table "PersonalInfo" (409 rows).
  - *AnswersNews.csv*: file with data from table "AnswerNews" (14,724 rows).
  - *News.csv*: file with data from table "News" (36 rows).
- *README.md*: Read me file with the dataset description and relevant information.
- *Questionnaire.pdf*: A document with the applied questionnaire, with all questions and possible answers.

In case *ReactSet* dataset (or part of it) is used for scientific, industrial, and/or academic purposes, or in case the dataset is publicly mentioned for whatever purposes, we request the acknowledgment of the authors by citing this work.

5. Conclusion

The study of emotions has attracted the attention of the scientific community for decades now. With the advance of social networks, many studies have been proposed to analyze users' behavior regarding different content. In this work, we approached this issue by providing a structured data source, collected from Facebook users. We presented *ReactSet*, a dataset with the reactions of users regarding different news. *ReactSet* gathers personal information from users, such as sex, age, and the number of hours it spends on Facebook. The dataset contains the information of 36 news, along with the number of reactions (Like, Love, Haha, Wow, Sad, Angry) it has received up to July 2019. For each news, we obtained reactions, classifications (positive, neutral, negative), and emotions (anger, disgust, fear, joy, sadness, and surprise) reported by users, considering both the image and the title of the news. We also provide the time the user spent to answer the questions related to each news, and the total time users took to answer the entire questionnaire.

The news selection went through a rigorous process involving postgraduate specialists in the area of Behavior Analysis, in order to search for the most emotionally significant images. The level of agreement among users was also evaluated. It is worth noting that we do not address the news that could be directed to a single, specific audience presenting or ideological or political bias, such as news of famous people or candidates for political offices. Although Facebook was the platform employed for data gathering, the reactions are expressions of the basic emotions defined by Ekman [Ekman et al. 1969, Ekman 1993] as shown in Figure 5. Accordingly, and without loss of generalization, *ReactSet* can be further expanded and explored.

We showed different data visualizations and application challenges to emphasize potential application scenarios for future analysis of *ReactSet*, which includes aspects of temporality, missing data, and recognition of sophisticated feelings and emotions. *ReactSet* is publicly available for download, under the Creative Commons license, and can be downloaded at *https://github.com/mtcazzolato/reactset*. We reinforce that all gathered information was obtained under the approval of the Ethical Committee of our institution, also respecting and preserving the identity of the participants of our study.

Acknowledgments

This research was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, by the São Paulo Research Foundation – FAPESP [grants #2018/24414-2, #2016/17078-0, #2019/01406-7], the National Council for Scientific and Technological Development (CNPq), and by the Centro de Ciências Matemáticas Aplicadas à Indústria (CeMEAI) [FAPESP grant #2013/07375-0].

Ethical Approval

This work has the approval of the Ethics Committee from the School of Arts, Sciences and Humanities (EACH) of the University of São Paulo, Brazil, under Register Number 88799118.8.0000.5390.

References

DeMiglio, L. and Williams, A. (2016). A sense of place, a sense of well-being. In *Sense of place, health and quality of life*, pages 35–50. Routledge.

Ekman, P. (1993). Facial expression and emotion. *American psychologist*, 48(4):384.

Ekman, P. and Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124.

Ekman, P., Sorenson, E. R., and Friesen, W. V. (1969). Pan-cultural elements in facial displays of emotion. *Science*, 164(3875):86–88.

Faraway, J. J. and Augustin, N. H. (2018). When small data beats big data. *Statistics & Probability Letters*, 136:142 – 145. The role of Statistics in the era of big data.

Fridlund, A. J. (1994). *Human facial expression: An evolutionary view*. Academic Press.

Frijda, N. H., Manstead, A. S., and Bem, S. (2000). *Emotions and beliefs: How feelings influence thoughts*. Cambridge University Press.

Gaspar, R., Pedro, C., Panagiotopoulos, P., and Seibt, B. (2016). Beyond positive or negative: Qualitative sentiment analysis of social media reactions to unexpected stressful events. *Computers in Human Behavior*, 56:179 – 191.

Giuntini, F. T., Ruiz, L. P., Kirchner, L. D. F., Passarelli, D. A., dos Reis, M. J. D., Campbell, A. T., and Ueyama, J. (2019). How do I feel? Identifying emotional expressions on facebook reactions using clustering mechanism. *IEEE Access*, 7:53909–53921.

Huang, A. H., Yen, D. C., and Zhang, X. (2008). Exploring the potential effects of emoticons. *Inf. Manage.*, 45(7):466–473.

Jibril, T. A. and Abdullah, M. H. (2013). Relevance of emoticons in computer-mediated communication contexts: An overview. *Asian Social Science*, 9(4):201.

Lerner, J. S. and Keltner, D. (2001). Fear, anger, and risk. *Journal of personality and social psychology*, 81(1):146.

Luor, T. T., ling Wu, L., Lu, H.-P., and Tao, Y.-H. (2010). The effect of emoticons in simplex and complex task-oriented communication: An empirical study of instant messaging. *Computers in Human Behavior*, 26(5):889 – 895.

Oleszkiewicz, A., Karwowski, M., Pisanski, K., Sorokowski, P., Sobrado, B., and Sorokowska, A. (2017). Who uses emoticons? data from 86702 facebook users. *Personality and Individual Differences*, 119:289 – 295.

Russell, J. A., Bachorowski, J.-A., and Fernández-Dols, J.-M. (2003). Facial and vocal expressions of emotion. *Annual Review of Psychology*, 54(1):329–349. PMID: 12415074.

Scherer, K. R. (1987). Toward a dynamic theory of emotion. *Geneva studies in Emotion*, 1:1–96.

Spoor, J. R. and Kelly, J. R. (2004). The evolutionary significance of affect in groups: Communication and group bonding. *Group Processes & Intergroup Relations*, 7(4):398–412.

Statista (2019). Number of monthly active Facebook users worldwide as of 1st quarter 2019 (in millions). https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/. Acessed: 2019-07-7.

Vashisht, G. and Thakur, S. (2014). Facebook as a corpus for emoticons-based sentiment analysis. *Int. J. Emerg. Technol. Advan. Eng*, 4:904–908.

Waller, B. M., Cray Jr, J. J., and Burrows, A. M. (2008). Selection for universal facial emotion. *Emotion*, 8(3):435.

Wang, H. and Castanon, J. A. (2015). Sentiment expression via emoticons on social media. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2404–2408.

Wegrzyn-Wolska, K., Bougueroua, L., Yu, H., and Zhong, J. (2016). Explore the effects of emoticons on twitter sentiment analysis. *Computer Science & Information Technology*, page 65.

Wolf, K. (2015). Measuring facial expression of emotion. *Dialogues in clinical neuroscience*, 17(4):457.

G-FranC: A dataset of Criminal Activities mapped as a Complex Network in a Relational DBMS

Lucas Scabora[1], Gabriel Spadon[1], Lucas S. Rodrigues[1], Mirela T. Cazzolato[1], Marcus V. S. Araujo[1], Elaine P. M. Sousa[1], Agma J. M. Traina[1] Jose F. Rodrigues-Jr[1], Caetano Traina-Jr[1]

[1]Institute of Mathematics and Computer Sciences – USP – São Carlos, Brazil

{lucascsb,spadon,lucas_rodrigues,mirelac,araujo}@usp.br

{parros,agma,junio,caetano}@icmc.usp.br

Abstract. *This work presents G-FranC, a dataset that combines the structure of a complex network of urban streets and criminal activities from the city of San Francisco – CA, USA, occurred from the year 2003 to 2018. We describe the carried preprocessing steps, related application scenarios, and challenges of the dataset. The combination of both crime data and the network topology enables diverse analysis, including the study of the unlawful activities within the city, their respective communities, and their spatial tendency along the years. G-FranC contributes with the combination of both data types in a unified and structured schema, stored in the PostgreSQL. G-FranC is available online, together with the script files to create and handle updates in the source data.*

1. Introduction

Complex networks can model data from a range of domains, such as networks of social interaction, product recommendation, communication infrastructure, and urban street organization [Barabási 2016]. In the context of urban systems, these networks describe streets of cities, traits related to transportation, and even the dynamics of collective human behavior [Kang et al. 2013], a resource that is amplified when the network is provided with additional data, such as urban indicators [Porta et al. 2009]. Crime data is one among many existing urban indicators. It describes a particular type of human behavior that relates to unlawful activities from minor offenses to felonies. Appraising criminal activities is not an easy task and has long been studied on the related literature [Bettencourt et al. 2010, Fitterer et al. 2014, Bogomolov et al. 2014, Deryol et al. 2016]. Studies on this area have the potential to improve safety in cities by aiding in decision-making activities related to public policies planning. Joining information from both crime data and the network topology has the potential to describe which regions of cities are more crime-critical and require intervention [Deryol et al. 2016].

In this context, in [Nieto-Chaupis 2018a, Nieto-Chaupis 2018b] the authors reviewed aspects of social duality regarding indices of street criminality together with traffic data from Lima – Peru, and studied the aspects related to the identification of social anomalies in a range of Latin American cities, respectively. The authors of [Ferreira et al. 2018] applied complex-network techniques to understand characteristics related to public security in Bogota – Colombia. In [Spadon et al. 2016] the authors focused on the task of evaluating criminal regions within a city, reviewing cases

in which similar crimes tend to concentrate in adjacent areas. In [Spadon et al. 2017] the authors proposed an algorithm to measure the criminality dispersion in highly criminal regions, allowing the identification not only of regions with recurrent criminal activity but also particularities related to the criminality spread within different regions of the same city. Other examples of works from literature include [Galbrun et al. 2016], which focused on crime mapping to estimate the probability of crime occurrences within street segments, [Fitterer et al. 2014] which employed statistical models to predict crimes, and [Bogomolov et al. 2014] which used mobile phone data to predict crime hot-spots.

Studies based on complex networks (also referred to as graphs) are structured in terms of two sets, the nodes (or entities) and links (or relationships), which can be efficiently stored and managed through relations and tables from Relational Database Management Systems (RDBMSs). RDBMSs can be used to aid in the manipulation of these relations and tables among various users and applications [Elmasri and Navathe 2015]. However, the related literature still lacks on approaches to manage both complex networks and crime data stored into a single RDBMS schema.

This work provides a dataset that combines the structure of a complex network of urban streets and criminal activities within a unified and structured schema stored in an RDBMS. More specifically, our contributions are in the Extraction, Transformation, and Loading (ETL) data processes [Kimball and Caserta 2011], combining both data sources into a single RDBMS schema. As a result, we created the dataset named `G-FranC`, which merges the structural data of the city of San Francisco – CA, USA, together with criminal activities from all over the city. `G-FranC` contains loading scripts that facilitates the management of these data, in Structured Query Language (SQL), which are straightforward compatible with the RDBMS PostgreSQL. Once `G-FranC` is loaded, several systems or programming languages that connect to the PostgreSQL can access the dataset to perform studies and analysis. Aiming for maintainability, we also provide all the scripts used in the ETL process to generate the `G-FranC` dataset and to cover future updates, in order to include new crime occurrences or changes in the San Francisco graph.

The remainder of this paper is organized as follows. Section 2 presents a detailed description of `G-FranC`, including the data acquisition (Section 2.1), crime mapping (Section 2.2), community detection (Section 2.3), and database schema creation (Section 2.4). Section 3 exemplifies three analyses over `G-FranC`, illustrating its potential. Section 4 discusses potential applications and the main challenges where `G-FranC` can be employed. Section 5 provides information on how to obtain our dataset and to use the additional code scripts. Finally, Section 6 presents the conclusions.

2. `G-FranC`: Dataset assembling

The pipeline used to create the dataset (see Figure 1) receives as input the criminal activity (Crimes Data) and complex network of the city of San Francisco – CA, USA (OSMnx Data). It is followed by the preprocessing of the nodes, links, and crimes, which will generate the scripts used for loading the dataset into the PostgreSQL. Lastly, it explores the task of community detection using the Nerstrand algorithm, which enables the discovery of criminal communities within cities. Following, we detail each of these activities.

2.1. Data acquisition

`G-FranC` combines data from different sources, described in the following.

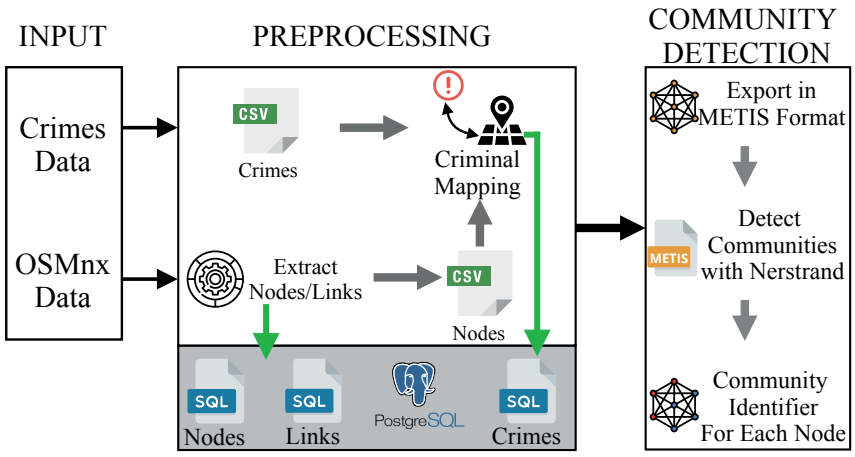


Figure 1. G-FranC assembly process showing the steps of data acquisition, mapping of criminal activities, and discovering of criminal communities.

Georeferenced graph source. We refer to a georeferenced graph as a directed graph $G = \{V, E\}$, in which E is the set of street segments (*i.e.*, links) and V is the set of streets intersections (*i.e.*, nodes). Each node $v \in V$ has at least one identifier (*i.e.*, id) and a set of georeferenced coordinates (*i.e.*, latitude and longitude). Each link $e \in E$ corresponds to an ordered pair $\langle s, t \rangle$, with $s \in V$ and $t \in V$, where s is referred to as *source* and t as *target*. It is worth mentioning that, in the context of this paper, the georeferenced graph corresponds to a complex network describing the street mesh of the city of San Francisco – CA, USA, which was acquired through the OSMnx[1] Python package [Boeing 2017].

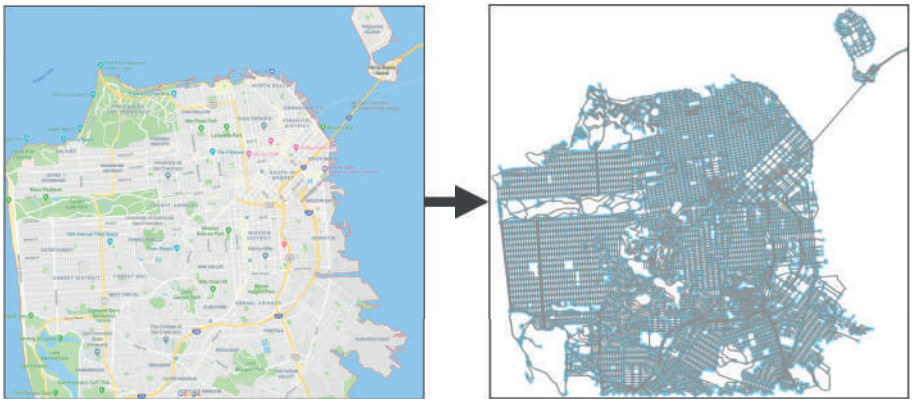


Figure 2. Image of San Francisco – CA, USA, extracted from Google Maps (on the left-hand side) and created using OSMnx (on the right-hand side).

Figure 2 shows the San Francisco graph data extracted by OSMnx[2], with $|V| = 9,559$ nodes and $|E| = 26,817$ links. Each node comes with an identifier that ranges from $32,927,563$ to $6,522,764,213$. However, as a criterion of the community detection algorithm (described in Section 2.3), we added a sequential identifier to each node, from 1 to $|V|$. Regarding the links, each $e \in E$ has an attribute that corresponds to the length (or distance) of the street segment (in meters). Notice that, the OSMnx graph

[1]Available at `https://github.com/gboeing/osmnx`.
[2]San Francisco nodes and links were extracted in July 1st, 2019.

comes as a multi-graph object by default, meaning that the resulting graph has multiple links connecting the same pair $\langle s, t \rangle$ of nodes, with different values of length. To ensure uniqueness, we filtered out the data to maintain only the link with the highest street segment length among all repeated links across the nodes. After filtering out the 128 duplicated links, the final number of links is $|E| = 26,689$.

We created a script in Python that builds and prepare the SQL script that creates and populates the *Node* and *Link* tables, as detailed in Section 2.4. The Python script also generates a Comma-Separated Values (CSV) file with all network nodes, which will be used in the crime mapping step, detailed in Section 2.2.

Criminal activity source. The criminal activity dataset C is provided by *San Francisco OpenData* (SFO) initiative[3] and includes incident reports made by police officers and by individuals (*i.e.*, citizens). Currently, there are a total of $2,215,024$ crimes that correspond to the period between January 1st, 2003 through May 15th, 2018. The main attributes are the crime category and spatial coordinates. Table 1 shows the 7 most frequent crime categories, from the total of 39 crime categories, present in `G-FranC`.

Table 1. The 7 most frequent crime categories among the dataset, including the reported crime category, number of occurrences, and an example description.

#	Category	Occurrences	Examples
1	Larceny/Theft	480,420	Theft from private properties and shoplifting.
2	Other Offenses	309,320	Traffic violation and resisting arrest.
3	Non-criminal	238,313	Violation of civil sidewalks and fire or death reports.
4	Assault	194,685	Threats against life and physical child abuse.
5	Vehicle Theft	126,587	Reports about a stolen automobile, truck, or motorcycle.
6	Drug/Narcotic	119,621	Narcotics possession and marijuana transportation.
7	Vandalism	116,058	Malicious mischief.

2.2. Mapping crime data to a georeferenced graph

Preprocessing the criminal activity includes mapping the crimes to the network nodes (*i.e.*, street intersections). We discarded the invalid coordinates by filtering only the crimes whose latitude is between 37.0 and 38.0, and longitude is between -121.0 and -123.0, which corresponds to the metropolitan area of San Francisco. Then, we filtered out 143 crimes with incorrect coordinates, resulting in a total of $2,214,881$ valid crimes.

For each crime $c \in C$, we executed a K-Nearest Neighbors (KNN) query to map the crime to the closest node using $k = 1$. The distance function employed was the great-circle distance ($\mathfrak{D}$), calculated over the coordinates between every element $c \in C$ and $v \in V$. This distance corresponds to the real length between georeferenced elements, and is calculated over the Earth's surface. In other words, $\mathfrak{D}_{cv}$ denotes the great-circle distance between the crime $c \in C$ and the node $v \in V$, and is formally defined as:

$$\mathfrak{D}_{cv} = \mathcal{R} \times cos^{-1}(\ sin(l^c_{at})sin(l^v_{at}) \ + cos(l^c_{at})cos(l^v_{at})cos(\triangle^{lon}_{cv}) \ ) \quad (1)$$

[3]Available at `https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-Historical-2003/tmnf-yvry`.

where l^c_{at} and l^v_{at} are the latitude values, $\triangle^{lon}_{cv}$ is the difference between the longitude values l^c_{on} and l^v_{on}, and $\mathcal{R}$ is the earth's radius (*i.e.*, $6,371$km). Notice that, this distance is derived from the Spherical Law of Cosines, with all values represented in radians [Wylie 2011].

We created a code script that performs both filtering and mapping of valid crimes in C++ programming language. Such a script generates an SQL script to create and populate the Table *Crime*, as detailed in Section 2.4.

2.3. Detecting criminal communities

After mapping every crime $c \in C$ to the closest node $v \in V$, the final step to generate `G-FranC` is to detect criminal communities. We created a procedure using Procedural Language/PostgreSQL (PL/pgSQL)[4] to export the graph data to METIS[5] format. Using the METIS file as input, the process proceeds by executing the Nerstrand[6] algorithm, a multi-thread cluster detection tool-set for discovering clusters in graph-like data by maximizing the clusters' modularity [LaSalle and Karypis 2015]. Figure 3 shows an example of the process of community detection through Nerstrand in `G-FranC`.

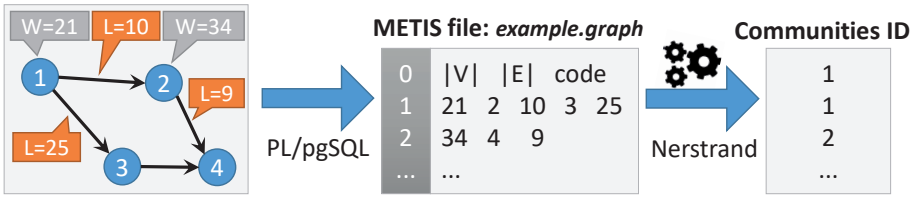


Figure 3. Criminal community detection using Nerstrand in `G-FranC`.

Here, we consider a graph with weights assigned to each node ("W" in Figure 3) and link ("L" in Figure 3). The node's weight refers to the number of crimes mapped to the node, and the link's weight refers to the street segment length. In this step, we cover four different scenarios: the graph (i) without crimes (*i.e.*, all weights equals to 0) and the ones with the 3 most frequent crimes (*i.e.*, (ii) Larceny/Theft, (iii) Other Offenses and (iv) Non-Criminal), resulting in four different graphs. In the case of multiple crime mapping, a different METIS file is generated for each crime category (see Table 1).

The METIS file is delimited by space characters. The first line of the file refers to the header, containing the number of nodes ($|V|$) and links ($|E|$), and a code. The code field determines whether the graph has weights attached to its nodes or links. Whenever there are weights in the nodes, an additional integer must be provided to inform the number of weights. Accordingly, we use nodes and link weights in `G-FranC`, and a single node weight, corresponding to code number "11 1". The remaining lines refer to the nodes, and each line number corresponds to the node identifier. The format expected for each node is the nodes' weight and a list of destination nodes with their respective weights (as illustrated in Figure 3). The `G-FranC` dataset provides not only the communities for the four aforementioned scenarios, but also the procedure to create the METIS file.

[4]Available at `https://www.postgresql.org/docs/current/plpgsql.html`.
[5]Available at `https://people.sc.fsu.edu/~jburkardt/data/metis_graph/`.
[6]Available at `https://github.com/dlasalle/nerstrand`.

2.4. Dataset schema

After the steps of preprocessing, mapping crimes and generating criminal communities, the three Tables *Node*, *Link* and *Crime* of `G-FranC` were automatically created and populated. Figure 4 shows a detailed view of the database schema.

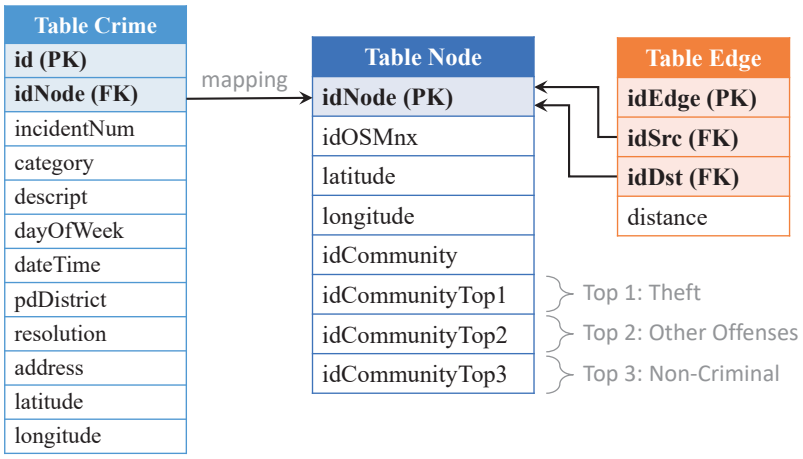


Figure 4. `G-FranC`'s Schema.

Table *Node*. It stores street intersections (*i.e.*, nodes), and has 8 attributes: (i) *idNode*: the sequential node identifier used as the primary key (*i.e.*, PK in the figure); (ii) *idOSMnx*: the node identifier used by the OSMnx; and, (iii,iv) *latitude* and *longitude*: the coordinates. The four remaining attributes refer to the discovered communities (see Section 2.3), with no crime data (*i.e.*, attribute *idCommunity*), and with the 3 most frequent crimes: Larceny/Theft (in attribute *idCommunityTop1*), Other Offenses (in attribute *idCommunityTop2*) and Non-Criminal (in attribute *idCommunityTop3*). We employed the Nestrand algorithm with default hyperparameters to discover all the communities.

Table *Link*. It stores street segments (*i.e.*, links) and has 4 attributes: (i) *idLink*: the sequential link identifier used as the primary key; (ii) *idSrc*: the identifier of the *source* node; (iii) *idTgt*: the identifier of the *target* node; and, (iv) *length*: the length of the street segment, in meters. Both attributes *idSrc* and *idTgt* are defined as foreign keys (*i.e.*, FK in the figure) to the table *Node*. The attribute *length* ranges from 0.134 to 3232.712 meters.

Table *Crime*. It stores crime events and has 12 attributes: (i) *id*: the sequential crime used as the primary key; (ii) *idNode*: the node identifier FK, which is the closest node to the crime (see Section 2.2); (iii) *incidentNum*: the incident identifier from the SFO initiative; (iv) *category*: the crime category; (v) *descript*: the description of the category; (vi) *dayOfWeek*: the weekday when the crime occurred (*e.g.*, Monday); (vii) *dateTime*: the timestamp of when the crime occurred; (viii) *pdDistrict*: the district where the crime occurred (*e.g.*, Bayview or Richmond); (ix) *resolution*: the resolution (if exists) of the crime (*e.g.*, "ARREST, BOOKED", that means that the subject was arrested); (x) *address*: the address where the crime occurred; and, (xi,xii) *latitude* and *longitude*: the coordinates.

3. Dataset analysis examples

We provide three analysis to illustrate the potential of the G-FranC dataset: (i) Neighborhood Analysis: the proportion of crime categories from node neighborhoods; (ii) Community Analysis: the spatial evaluation of the 5 largest criminal communities; and, (iii) Temporal Analysis: some criminal tendencies considering their temporal information.

Neighborhood analysis. This section exemplifies the analysis of neighborhood queries that can be carried out on G-FranC. Figure 5 shows a neighborhood query performed over the node with the highest number of crime occurrences (red star), and nodes within the ranges of 0.5 km (blue squares) and 1 km (green circles) using the sum of the street segment lengths. We show a chart for each query, with the proportion of the 7 most frequent crime categories (according to Table 1). We observe variations in the proportions of some crime categories, such as Other Offenses and Vehicle Theft, which are smaller in the node with the highest number of crimes than within the nodes in the neighborhood. Such observation can result in various analysis, for instance, the analysis of similarity among crime categories both between specific nodes and its neighborhood regions.

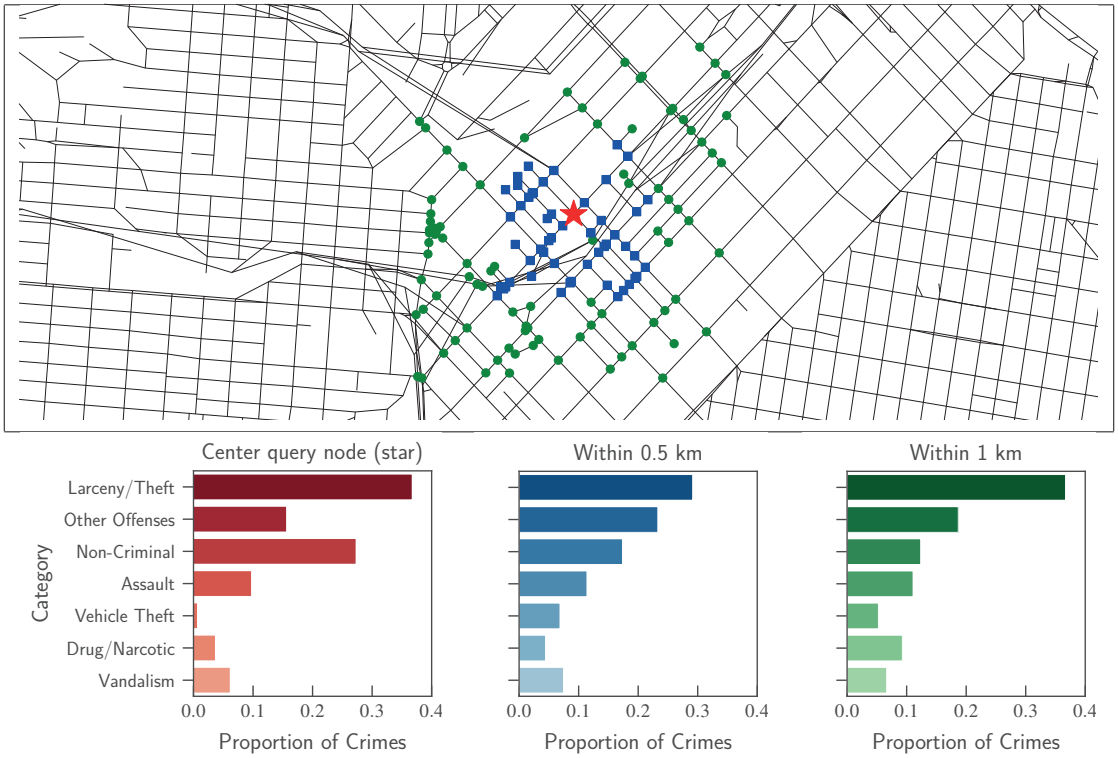


Figure 5. Neighborhood analysis of the node with the highest number of crime occurrences (red star), the nearest nodes within 0.5 km (blue squares) and 1 km (green circles). Additionally, we provide the proportion of the 7 most recurrent crime categories for each scenario.

Community analysis. For the community detection evaluation, we analyzed the overlapping areas of the 5 largest criminal communities for the categories Larceny/Theft and Other Offenses. Figure 6 shows these areas, with purple being the areas for Larceny/Theft,

orange being the areas for Other Offenses and red being their overlapping areas. The overlapping areas depict that these regions in the northeast and south of the city require special attention when evaluating a possible relation between these crime categories. Note that the analysis can be extended to several others, such as comparing the combinations of the 39 crime categories or evaluating the criminal density in these areas.

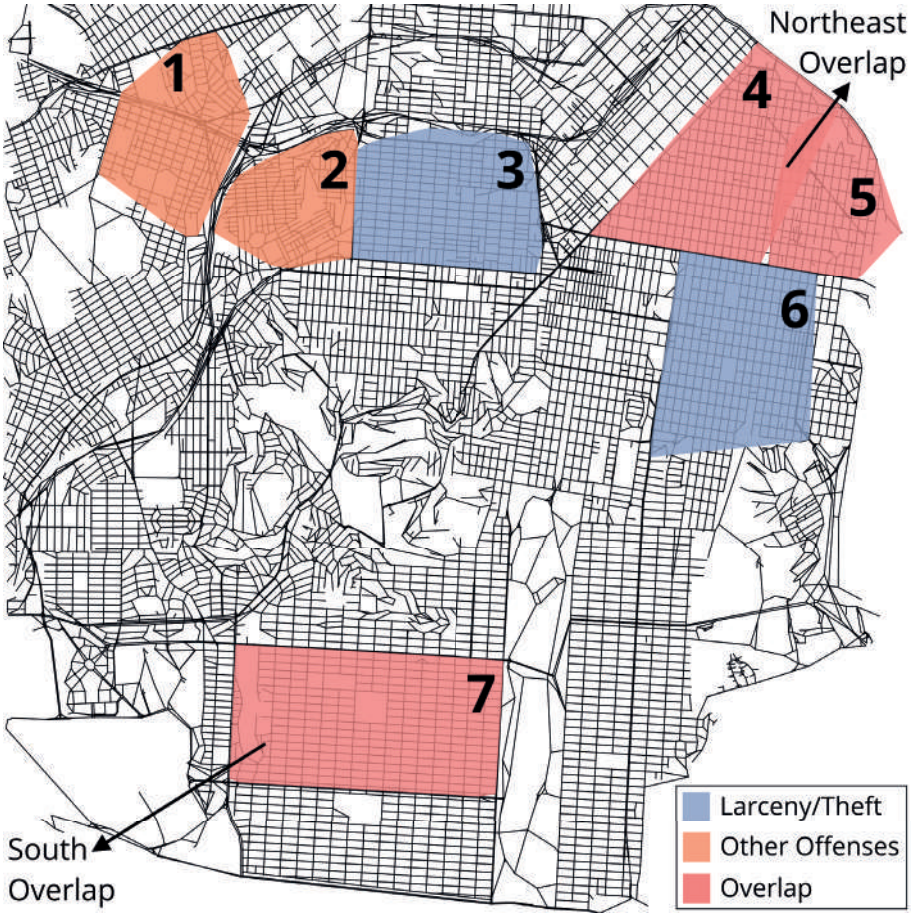
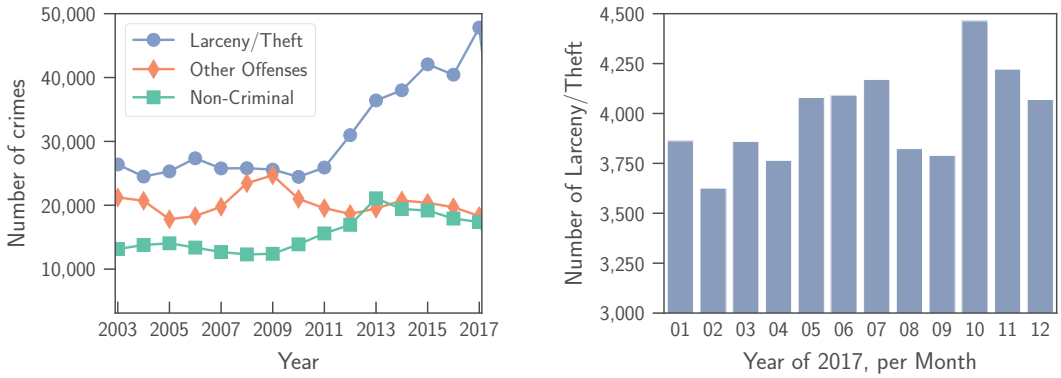


Figure 6. The 5 largest criminal communities considering Larceny/Theft (regions 3 to 7), Other Offenses (regions 1, 2, 4, 5 and 7), and their overlapping areas (regions 4, 5 and 7).

Temporal analysis. Finally, a not yet explored possibility in `G-FranC` is to consider crimes with timestamps to perform a temporal-based analysis. Figure 7(a) illustrates the time series generated for the 3 most frequent crime categories. Here we notice that the Larceny/Theft keeps increasing along the years, while Other Offenses and Non-Criminal have different behavior. With a total of 39 crime categories, it is possible to compare these temporal behaviors aimed at discovering correlations, predictions, or both. Figure 7(b) details the number of crimes of Larceny/Theft reported in 2017, which is the year with the highest number of reports. This figure shows that August and September are uncommonly smaller than their previous and subsequent months (*i.e.*, June and October). Such odd behavior may be explored by analyzing other years (considering a possible seasonality in such data) and comparing the tendency of different crime categories (*e.g.*, Assault, Drug/Narcotic and Vandalism from Table 1). `G-FranC` allows performing those analyses also in regional scope.



(a) Evaluation per Year. (b) Evaluation per Month by fixing a Year.

Figure 7. Examples of temporal analysis over `G-FranC` per year (a) and per month (b), considering their categories and timestamps.

4. Applicability and challenges for `G-FranC`

`G-FranC` can be employed in several analysis and applications. Following, we highlighted some scenarios and challenges where `G-FranC` can be employed.

Applicability. Since `G-FranC` is stored in an RDBMS, it is possible to access its data through several programming languages (*e.g.*, Python, C++, Java, Matlab, and R). For instance, all the examples of Section 3 were implemented in Python using the psycopg2 library[7] to load and manage the data. We provided all the scripts and codes to preprocess and generate the `G-FranC` dataset with recent data (by adapting to new crimes and keeping track of possible alterations in the city of San Francisco). Moreover, although `G-FranC` was employed in an RDBMS, it can be exported in CSV format and loaded into other systems, including graph-oriented NoSQL databases[8].

Challenges and Limitations. There are three main challenges in the processing of `G-FranC`. The first one refers to the evolution of the criminal communities along the time (*e.g.*, combining the analysis of Figures 6 and 7). This case refers to the analysis of increasing or decreasing (in the number of nodes and absolute size) the criminal communities identified by the Nerstrand algorithm. This analysis can contribute to measure the effectiveness of public policies related to security measures. The second challenge is related to the connection between distinct crimes in order to predict cause and effect relationships based on the location and type of crime. Despite difficult, this task is of interest to law enforcement, with applicability to trace relations among crimes on a global network level. The third challenge is related to including graph metrics into the criminal analysis, such as centrality metrics (*e.g.*, Closeness and Betweenness). These metrics have long been used to describe several phenomena observed in cities, and we believe such relations can be extended to crime data when relating the centrality of a node with the number of nearby crimes. `G-FranC`'s main limitation is the lack of crimes mapped to some nodes as there is no criminal activity surrounding it. This same phenomenon can be observed in cases where two nodes are adjacent to each other. In cases like this, it is common to see a crime mapped just to one of them. Since we allowed updates from new crime occurrences, such limitation tends to reduce with time.

[7] Available at `http://initd.org/psycopg/`.

[8] To load the CSV file in Neo4j see: `https://neo4j.com/developer/guide-import-csv/`.

5. Download and citation request

We made G-FranC and the related code scripts available for download at https://bitbucket.org/gbdi/g-franc/. The dataset is freely available for the public under the Creative Commons BY license[9]. When using G-FranC (or part of it) of any kind, we request acknowledgment by citing this paper.

The repository is structured as follows:

- File create_and_insert.zip has the SQL scripts that create the tables and insert the data of G-FranC into the RDBMS PostgreSQL; and,
- File codes.zip, with the code scripts to generate an updated version of G-FranC, and which is divided into: (i) script "*gathering.py*", a Python that extracts the graph data using the OSMnx package (see Section 2.1); (ii) folder "*mapping*", containing the code in C++ that preprocesses and maps the crime data to the closest node. We also provided a *Makefile* with examples of code compilation and execution (see Sections 2.1 and 2.2); and, (iii) script "*exportMETIS.sql*", that generates a METIS file for computing the criminal communities according to the provided crime category (see Section 2.3). This METIS file can be directly used as input to the Nerstrand program for criminal community detection.

6. Conclusion

In this work, we presented G-FranC, a dataset which combines the structure of a complex network of urban streets together with data describing criminal activities in a unified and structured schema. Our contributions are in the Extraction, Transformation, and Loading (ETL) processes, combining both data sources in a single RDBMS schema. Additionally, we provided three examples of analysis that can be carried out on G-FranC, highlighting some future possibilities, applications, and challenges. G-FranC is structured as a relational schema, and is available in a public repository under the Creative Commons license. Finally, aiming for maintainability, we also provide all the scripts used in the ETL process to generate G-FranC in order to cover source data updates.

Acknowledgments. The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001; Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), through grants 2016/17078-0, 2016/17330-1, 2017/08376-0, 2019/04461-9, and 2018/24414-2; and, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for their financial support.

References

Barabási, A.-L. (2016). *Network Science*. Cambridge University Press.

Bettencourt, L. M. A., Lobo, J., Strumsky, D., and West, G. B. (2010). Urban Scaling and Its Deviations: Revealing the Structure of Wealth, Innovation and Crime across Cities. *PLoS ONE*, 5(11):e13541.

Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139.

[9]Creative Commons BY 4.0 license: https://creativecommons.org/licenses/by/4.0/.

Bogomolov, A., Lepri, B., Staiano, J., Oliver, N., Pianesi, F., and Pentland, A. (2014). Once Upon a Crime. In *Proceedings of the 16th International Conference on Multimodal Interaction - ICMI '14*, volume 1, pages 427–434. ACM.

Deryol, R., Wilcox, P., Logan, M., and Wooldredge, J. (2016). Crime Places in Context: An Illustration of the Multilevel Nature of Hot Spot Development. *Journal of Quantitative Criminology*, 32(2):305–325.

Elmasri, R. and Navathe, S. B. (2015). *Fundamentals of Database Systems*. Pearson, 7th edition.

Ferreira, A., Rubiano, G., and Mojica-Nava, E. (2018). Urban Security Analysis in the City of Bogotá Using Complex Networks. In *International Conference on Complex Systems*, pages 424–438. Springer.

Fitterer, J., Nelson, T., and Nathoo, F. (2014). Predictive crime mapping. *Police Practice and Research*, 16(2):121–135.

Galbrun, E., Pelechrinis, K., and Terzi, E. (2016). Urban navigation beyond shortest route: The case of safe paths. *Information Systems*, 57:160–171.

Kang, C., Sobolevsky, S., Liu, Y., and Ratti, C. (2013). Exploring human movements in singapore: a comparative analysis based on mobile phone and taxicab usages. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*.

Kimball, R. and Caserta, J. (2011). *The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data*. John Wiley & Sons, USA.

LaSalle, D. and Karypis, G. (2015). Multi-threaded modularity based graph clustering using the multilevel paradigm. *J. Parallel Distrib. Comput.*, 76(C):66–80.

Nieto-Chaupis, H. (2018a). Identification of the Social Duality: Street Criminality and High Vehicle Traffic in Lima City by Using Artificial Intelligence Through the Fisher-Snedecor Statistics and Shannon's Entropy. In *IEEE International Smart Cities Conference (ISC2)*, pages 1–6.

Nieto-Chaupis, H. (2018b). Shannon-entropy-based artificial intelligence applied to identify social anomalies in large latin american cities. In *IEEE 39th Sarnoff Symposium*, pages 1–4.

Porta, S., Strano, E., Iacoviello, V., Messora, R., Latora, V., Cardillo, A., Wang, F., and Scellato, S. (2009). Street centrality and densities of retail and services in Bologna, Italy. *Environment and Planning B Planning and Design*, 36:1–15.

Spadon, G., Scabora, L. C., Araujo, M. V. S., Oliveira, P. H., Machado, B. B., de Sousa, E. P. M., Jr., C. T., and Jr., J. F. R. (2016). Complex network tools to understand the behavior of criminality in urban areas. *CoRR*, abs/1612.06115.

Spadon, G., Scabora, L. C., Oliveira, P. H., Araujo, M. V. S., Machado, B. B., de Sousa, E. P. M., Jr., C. T., and Jr., J. F. R. (2017). Behavioral characterization of criminality spread in cities. In *International Conference on Computational Science (ICCS), 12-14 June, Zurich, Switzerland*, pages 2537–2541.

Wylie, C. R. (2011). *Introduction to Projective Geometry*. Courier Corporation.

Índices de Infoboxes para Recuperação de Informação Estruturada de Entidades da Wikipédia

Johny Moreira[1], **Everaldo Costa Neto**[1], **Luciano Barbosa**[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brasil

{jms5, ecsn, luciano}@cin.ufpe.br

Abstract. *Wikipedia infoboxes have emerged as a valuable data source in the Web. Some works have used this data for different tasks. The DBPedia project provides the data in specific datasets. However, there is a trade-off related to data quality and coverage in these datasets. Thus, this work defines a process to directly extract infoboxes from Wikipedia Dump. The collected data have gone through a processing step and were indexed using Apache Lucene. The built indexes were made available and are ready to consume. The main contribution of this work is the extracted Wikipedia Infoboxes data, organized and provided for general use. The data is provided with preserved coverage and noise reduction.*

Resumo. *Infoboxes da Wikipédia têm emergido como uma valiosa fonte de dados estruturados na Web. Diversos trabalhos utilizam esses dados para as mais diferentes tarefas. A DBpedia disponibiliza esses dados em alguns datasets específicos, entretanto, há um trade-off com relação à qualidade e a cobertura dos dados contidos nesses datasets. Dessa forma, este trabalho especifica um processo para extrair os dados de infoboxes diretamente do dump da Wikipédia. Os dados coletados passaram por uma etapa de processamento e foram indexados utilizando o Apache Lucene. A principal contribuição desse trabalho é a extração de dados estruturados da Wikipédia (infoboxes), organizados e disponibilizados para uso geral. Os índices criados estão disponíveis para uso, preservando a cobertura e diminuindo a quantidade de ruídos nos dados.*

1. Introdução

A Wikipédia é uma enciclopédia colaborativa, universal e multilíngue estabelecida na internet sob o princípio *wiki*[1]. De maneira geral, a Wikipédia contém um conjunto de páginas que descrevem conteúdo sobre diferentes entidades (e.g. políticos, produtos, empresas, artistas). A estrutura básica de uma página é formada por dois principais elementos: um texto, comumente organizado em seções, e um infobox. Um infobox pode ser visto como um conjunto de pares de propriedade-valor que sumariza, de maneira estruturada, informações sobre uma entidade específica.

Infoboxes da Wikipédia têm emergido como uma valiosa fonte de dados estruturados na Web, uma vez que a facilidade de trabalhar com dados estruturados é maior do que com dados não estruturados. A disponibilidade desses dados têm aberto novas oportunidades em trabalhos de diferentes áreas em comunidades como as de Banco de Dados,

[1]É um website no qual utilizadores modificam colaborativamente conteúdo e estrutura diretamente do *web browser*.

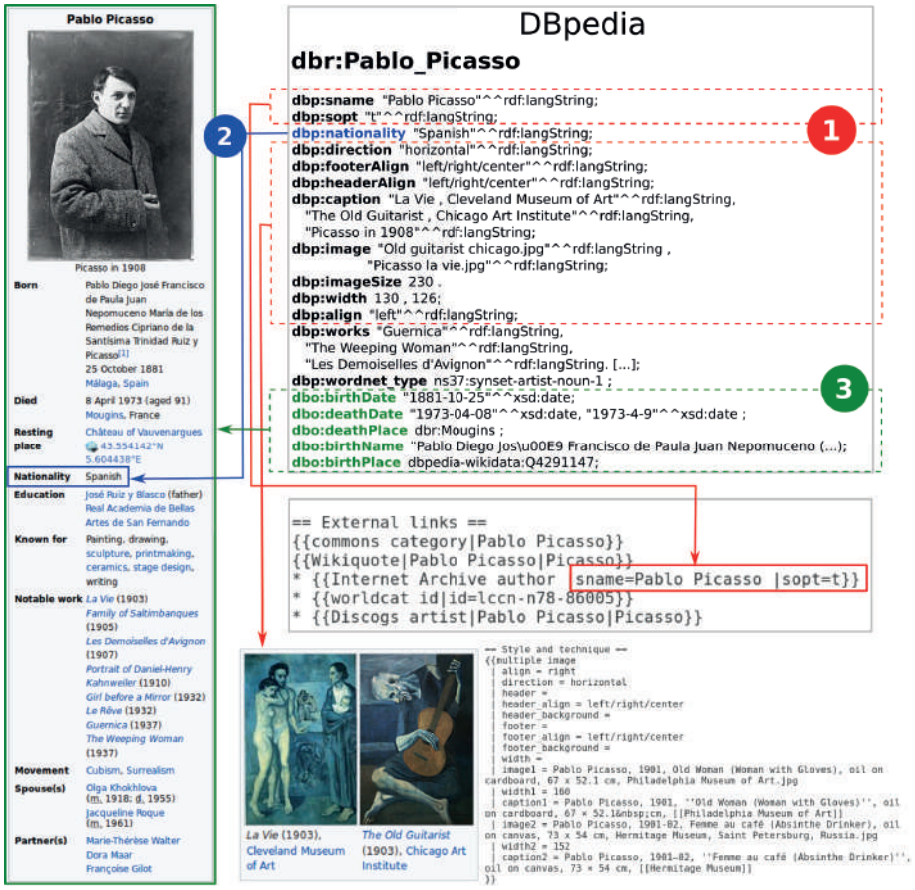


Figura 1. Dados do infobox da entidade Pablo_Picasso (esquerda); Amostra de dados contidos nos datasets da DBpedia (acima-direita); Trechos de *wikicode* (abaixo-direita).

Web Semântica, Processamento de Linguagem Natural, entre outras [Lange et al. 2010, Nguyen et al. 2011, Serra et al. 2011, Nguyen et al. 2012, Kuzey and Weikum 2012, Wecel and Lewoniewski 2015, Morales et al. 2016, Sáez and Hogan 2018].

O projeto DBpedia foi criado em 2007 com o objetivo de extrair conteúdo estruturado da Wikipédia. Extratores de informações são executados com o intuito de coletar, organizar e disponibilizar o conteúdo das páginas para consumo. Diferentes datasets são disponibilizados, cada um contendo informações específicas como, por exemplo, *Articles Categories*, que armazena o mapeamento entre páginas e categorias. A última versão da DBpedia é a 2016-10[2], lançada em 2017. Os dados são disponibilizados em formato de triplas RDF.

Especificamente, os dados de infobox estão contidos em alguns datasets, dentre os quais destaca-se: *Infobox Properties* e os *Mapping-based*[3]. Entretanto, há um *trade-off* com relação à qualidade e a cobertura dos dados contidos nesses datasets. Em *Infobox Properties* é possível encontrar muito ruído, uma vez que além dos dados de infoboxes os extratores consideram dados contidos em links externos e/ou quadro de informações que eventualmente a página possua. Um exemplo de uma situação semelhante pode ser vista

[2]https://wiki.dbpedia.org/downloads-2016-10
[3]Mappingbased Literals e Mappingbased Objects

na Figura 1.

O lado esquerdo da Figura 1 apresenta o infobox da página da entidade *Pablo_Picasso*[4]. O conteúdo apresentado no quadro superior do lado direito ilustra os dados encontrados nos, já previamente citados, datasets da DBpedia. As propriedades seguindo o namespace **dbp** (1) estão presentes no dataset *Infobox Properties*. Observe que algumas propriedades (*dbp:sname*, *dbp:sopt*, *dbp:direction*, *dpb:align*, entre outras destacadas) não possuem relação alguma com o infobox da entidade, elas são referentes à links externos, legendas e imagens presentes em quadros contidos na página da entidade. Esses ruídos são capturados pelos extratores responsáveis pela extração dos dados de infobox, adicionando informação desnecessária ou não relacionada diretamente com a entidade em questão. É importante destacar que devido a esses problemas de ruído e utilização de um namespace menos limpo, quando comparado ao namespace de ontologia, a utilização desse dataset não é recomendada pela própria comunidade DBpedia.

Em *Mappingbased Literals* e *Mappingbased Objects* os problemas anteriores são superados. A utilização de um namespace baseado na ontologia da DBpedia (**dbo**) torna esse dataset mais propício para uso. Porém, os extratores só consideram os infoboxes cujo template esteja mapeado para essa ontologia. Ocorre que muitos templates ainda não estão mapeados para a ontologia. Um exemplo disso é a página da entidade *Ghawar_Field*[5]. O template de infobox dessa entidade é o *Infobox_oilfield*. Esse template não está mapeado para ontologia, logo a entidade não apresentará seus repectivos dados no dataset em questão. Outro problema frequente é que, apesar de um template estar mapeado para ontologia (como é o caso de *Infobox_Artist*[6], usado pela entidade Pablo_Picasso), nem todas as suas propriedades estejam mapeadas para propriedade da ontologia. Uma situação como essa ocorre na Figura 1 (2); observe que a propriedade Nationality (*dbp:nationality*) aparece no infobox. Nos datasets *Mappingbased* essa propriedade não aparece para a entidade Pablo_Picasso, uma vez que a mesma não tem mapeamento definido, diferentemente das propriedades Died (*dbo:deathDate*), Born (*dbo:birthDate*), entre outras, identificadas com o prefixo dbo (3).

Diante da relevância desses dados para diferentes trabalhos, como os citados anteriormente, assim como para trabalhos posteriores, e das limitações encontradas nos datasets disponibilizados pela DBpedia, foi definido um processo para extrair infoboxes diretamente do *dump* da Wikipédia. Os dados coletados passaram por uma etapa de processamento e foram indexados no Apache Lucene[7]. Os índices gerados estão disponíveis para consumo. Como contribuição deste artigo tem-se os dados de infoboxes da Wikipédia extraídos, organizados e disponibilizados, com a cobertura preservada e livre de ruídos (contendo apenas dados de infoboxes).

O restante do trabalho está organizado como segue. Na Seção 2 é especificado o processo de coleta, processamento e indexação dos dados. Na Seção 3 é descrito como os dados podem ser acessados e consumidos. Na Seção 4 são apresentados cenários onde dados de infoboxes podem ser aplicados, discutindo alguns trabalhos do estado da arte. Por fim, na Seção 5 são realizadas algumas análises nos dados coletados e na Seção 6 são

[4] https://en.wikipedia.org/wiki/Pablo_Picasso
[5] https://en.wikipedia.org/wiki/Ghawar_Field
[6] http://mappings.dbpedia.org/index.php/Mapping_en:Infobox_artist
[7] https://lucene.apache.org/

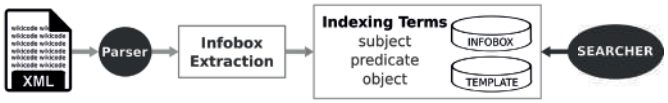


Figura 2. Visão geral do processo de extração e indexação dos dados de infoboxes

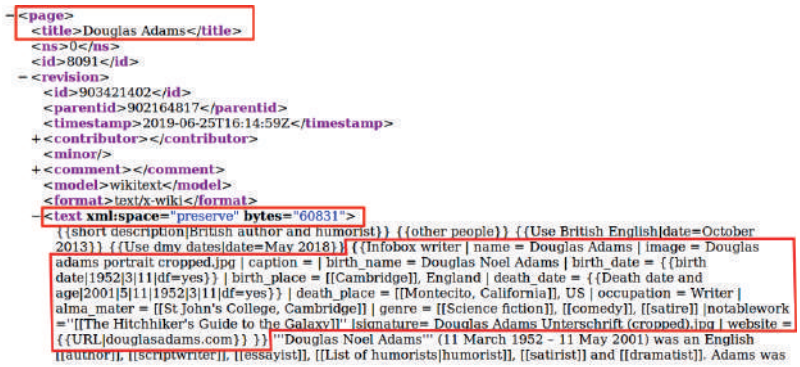


Figura 3. Amostra de dados do dump da Wikipédia

feitas as considerações finais.

2. Especificação do Processo

Os datasets referentes aos dados de infoboxes, disponibilizados pela DBpedia, apresentam algumas limitações específicas. Para superar essas limitações foi definido um processo que consiste em três etapas, cujo resultado final é a construção de dois índices: (i) infoboxes e (ii) templates. Esses índices permitem que consumidores interessados em dados de infoboxes da Wikipédia possam fazer uso desses dados para as mais diversas tarefas. A Figura 2 apresenta uma visão geral do processo.

2.1. Seleção dos Dados

O principal objetivo é extrair os dados de infoboxes de todas as entidades (páginas). Portanto, foi utilizado o *dump* completo da Wikipédia[8]. O referido arquivo está no formato XML e contém dados extraídos de todas as páginas da Wikipédia inglesa. A versão utilizada (10-2016), de acordo com o processo de extração desenvolvido, apresenta um total de 5.166.304 entidades (excluindo páginas administrativas), onde apenas 2.796.885 entidades possuem infobox. A Figura 3 ilustra como esses dados estão organizados no arquivo XML.

Cada entidade está associada em uma tag <page>, que por sua vez agrupa um conjunto de outras tags como: <title> (que identifica o nome da entidade) e <text> (que apresenta todo o conteúdo extraído da página). Como o dump possui um tamanho muito grande (aproximadamente 60 GBs) foi utilizado o *Apache Mahout*[9] para separar igualmente o dump XML sem corromper as suas marcações e assim realizar o processo de extração dos dados de forma paralela. A saída desta etapa é um conjunto de pares

[8] http://downloads.dbpedia.org/2016-10/core-i18n/en/pages_articles_en.xml.bz2

[9] https://mahout.apache.org/overview.html

Tabela 1. Regex para identificação de padrão de Infoboxes e sua proporção de uso

Padrão de Infobox	Proporção
{{\s?(I|i)nfobox.\n(|.*\n)*}}*	$88.49 \cdot 10^{-2}$
{{\s?(T|t)axobox.\n(|.*\n)*}}*	$9.48 \cdot 10^{-2}$
{{\s?(S|s)peciesbox.\n(|.*\n)*}}*	$52.72 \cdot 10^{-4}$
{{\s?(G|g)eobox.\n(|.*\n)*}}*	$52.23 \cdot 10^{-4}$
{{\s?(C|c)hembox.\n(|.*\n)*}}*	$37.71 \cdot 10^{-4}$
{{\s?(A|a)utomatic(_|\s)taxobox.\n(|.*\n)*}}*	$23.27 \cdot 10^{-4}$
{{\s?(D|d)rugbox.\n(|.*\n)*}}*	$22.47 \cdot 10^{-4}$
{{\s?(E|e)nzyme.\n(|.*\n)*}}*	$12.04 \cdot 10^{-4}$
{{\s?(P|p)fam(_|\s)?box.\n(|.*\n)*}}*	$2.06 \cdot 10^{-4}$
{{\s?(P|p)rotein.\n(|.*\n)*}}*	$37.89 \cdot 10^{-6}$

chave-valor, onde a chave é o nome da entidade e o valor é todo o texto contido na tag <text>.

2.2. Extração dos Dados (Parser)

É possível observar que a tag <text> inclui todo o conteúdo da página de uma entidade. Todavia, este trabalho está interessado apenas nos dados de infobox (parte demarcarda na Figura 3). Para que seja possível extrair apenas esse conteúdo, foi utilizado um *parser* extrai elementos do *wikicode* presente no dump. O *wikicode* pode ser definido como uma linguagem de marcação para formatação de páginas *wikis* pertencentes ao *MediaWiki Foundation*.

No *wikicode* estão incluídos *templates*[10], que podem ser identificados por meio de chaves duplas ({{ }}). Foi utilizado o *parser mwparserfromhell* [11] para gerar uma lista de templates e a partir dessa lista buscar por templates que representem infoboxes. Em uma análise prévia realizada nos *wikicodes* foi possível verificar que os templates de infoboxes seguem um padrão. Aproximadamente 90% desses templates iniciam com o termo "infobox", conforme mostra a Tabela 1. Outros templates que representam infoboxes mas que não seguem esse padrão de nomenclatura podem ser encontrados em menor proporção. Por essa razão, neste trabalho é realizada a extração de dados de infoboxes apenas de entidades que possuem instância de infoboxes cujo template inicia com o prefixo "infobox".

2.3. Indexação

Uma vez que o *mwparserfromhell* percorre todo arquivo em busca de dados de infobox, os dados extraídos são indexados no Apache Lucene. O Apache Lucene é uma API para indexação e busca de documentos. Foram criados dois índices: (i) infobox e (ii) template. Os dados foram indexados em três termos, *(<subject> <predicate> <object>)*, similar ao formato de triplas, utilizados nos datasets da DBpedia. A ideia de separar os dados em termos é para facilitar a busca, uma vez que separando a busca por termos é possível garantir que o resultado retornado manterá a integridade semântica. Por exemplo, suponha que haja interesse em recuperar todas as propriedades contidas no infobox da entidade *Pablo_Picasso*. Se fosse utilizado apenas um termo de indexação e toda a tripla RDF fosse armazenada nesse único termo, as triplas que contivessem o termo *Pablo_Picasso* como objeto também seriam incluídas no resultado da consulta. Essa organização de

[10] https://en.wikipedia.org/wiki/Help:Template
[11] https://mwparserfromhell.readthedocs.io/en/latest/

termos também foi utilizada a fim de manter um padrão de consulta e tornar mais intuitivo o consumo dos índices. A organização dos dados, em cada índice, pode ser vista abaixo.

infobox - Armazena a relação propridade-valor encontrada em uma instância de infobox. O termo *subject* é a entidade, o termo *predicate* é a propriedade e o termo *object* é o valor atribuido àquela propriedade. Exemplo: *<Pablo_Picasso>, <birth_place>, <Málaga, Spain>*.

template - Armazena a relação template-entidade encontrada na instância de infoboxes. Cada instância de infobox é associada à um template. Esse dado pode ser interessante pois permite identificar, por exemplo, quais entidades compartilham o mesmo template de infobox. Além disso, pode ser integrado com o outro índice para prover um resultado de consulta mais refinado. Neste índice, as tuplas estão organizadas da seguinte forma: O termo *subject* é a entidade, o termo *predicate* é vazio, e o termo *object* é o nome do template de infobox. Exemplo: *<Pablo_Picasso>, <null>, <Infobox_artist>*.

3. Acesso aos Índices

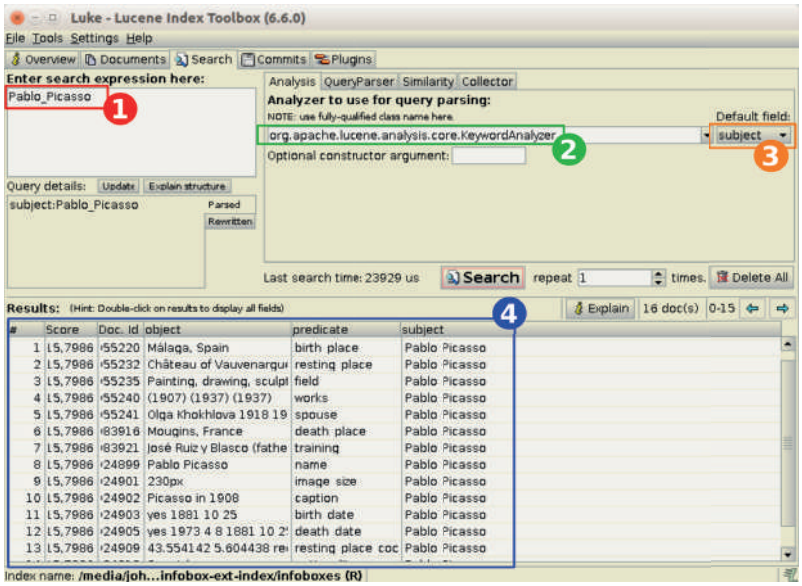


Figura 4. Consulta ao indice Infobox no Luke Lucene Index Toolbox

Os índices criados estão disponíveis em um repositório[12]. Os dados estão disponíveis em forma de índice por conta de diversos fatores, dentre eles: (i) o **volume de dados** coletados tem um tamanho de aproximadademente 3 GBs; (ii) a disponibilização desses dados em um formato de arquivo convencional (e.g. csv, json, xml) pode tornar o consumo dos dados mais lento, dependendo do recurso computacional disponível, diferentemente, o índice possui **mecanismos de busca e consumo mais eficientes**; (iii) o índice possibilita a **flexibilização de consultas específicas** que podem ser realizadas pelos consumidores de dados; (iv) o índice permite também que **o resultado de uma consulta possa ser exportado para outros formatos**, possibilitando a derivação de subconjuntos de dados específicos que podem ser consumidos/manipulados por outras aplicações.

[12] https://cin.ufpe.br/~jms5/infobox-ext-index.zip

```
public static void main(String[] args)
                        throws IOException, Exception {
    QueryData qData = new QueryData();
    Entity entity = qData.queryInfobox("Pablo_Picasso");

    // gerando visualizacao dos dados consultados
    InfoboxSchema infoboxInstance = entity.getInfobox();

    ... entity.getEntityTitle(); // entity name
    ... infoboxInstance.getTemplateName();  // template name

    for (InfoboxTuple tuple : infoboxInstance.getTuples()){
        ... tuple.toString();  // print infobox tuple
    }
}
```

Código 1. Exemplo de código executando uma consulta no Buscador utilizando a classe *QueryData*.

Para ilustrar como os dados disponibilizados podem ser consumidos suponha que haja interesse em *buscar pelo infobox da entidade Pablo_Picasso*. No índice os dados podem ser consumidos de duas formas: (i) Luke Lucene Index Toolbox (v. 6.6.0)[13] e (ii) API Apache Lucene. O Luke tem uma interface amigável e a definição de uma consulta é feita de forma mais simples, entretanto ele funciona como um visualizador de dados. A Figura 4 ilustra o resultado da consulta anterior. Em (1) é definido o termo de busca, em (2) é definido o analizador utilizado[14], em (3) é definido o termo de busca e em (4) é apresentado o resultado da consulta.

Outra forma de consumir os dados é utilizando a API Apache Lucene (versão 6.6.2) integrada ao código da aplicação. Para facilitar o consumo via API foi disponibilizado um **buscador**[15], que através da classe *Searcher* possibilita a consulta direta nos índices disponibilizados. Ainda no buscador existe a classe *QueryData*, que apresenta implementações de consultas específicas utilizando os métodos disponibilizados pela classe *Searcher*. Dentre os métodos existentes destaca-se:

- **queryAllInfoboxes()** - retorna uma lista de entidades;
- **queryEntitiesUsingTemplate(java.lang.String templateName)** - recebe como parâmetro um template de infobox e retorna uma lista de entidades que utiliza esse template;
- **queryInfobox(java.lang.String entityTitle)** - recebe como parâmetro uma entidade e retorna o infobox dessa entidade.

A documentação completa referente as classes e métodos do buscador está disponível para acesso[16]. É importante deixar claro que outras consultas podem ser especificadas, caso o usuário deseje. No código 1 é apresentado um exemplo de como a consulta que definimos acima pode ser feita utilizando a classe *QueryData*.

4. Aplicações e Trabalhos Relacionados

Na literatura é possível encontrar trabalhos relevantes para diferentes comunidades (e.g. Banco de Dados, Web Semântica, Processamento de Linguagem Natural, entre outras)

[13] https://github.com/DmitryKey/luke/releases/tag/luke-6.6.0

[14] Analisadores no Lucene são uma junção de tokenizador, *stemmers* e filtros de *stop-words*. Como a busca aqui realizada necessita de combinação exata, foi optado pela utilização do analisador *KeywordAnalyzer* que não realiza qualquer modificação nos campos de indexação e busca.

[15] https://github.com/guardiaum/InfoboxIndexSearch

[16] https://guardiaum.github.io/InfoboxIndexSearch/index.html

que fazem uso de dados de infoboxes para realizar tarefas específicas. Nesta seção alguns desses trabalhos são agrupados por tarefas, com o objetivo de corroborar a relevância dos dados de infoboxes da Wikipédia.

Busca Estruturada em Documentos - Um documento da Wikipédia é formado por um texto (dado não estruturado) e um infobox (dado estruturado). O mapeamento entre esses dois tipos de dados permite que buscas mais complexas, para recuperação de documentos não estruturados, possam ser realizadas. Os trabalhos de [Nguyen et al. 2011, Nguyen et al. 2012] estão inseridos nesse contexto, mais especificamente nas tarefas de mapeamento de infoboxes. Em [Nguyen et al. 2011] os autores utilizam dados de infobox para encontrar, de forma automática, mapeamentos entre propriedades de diferentes línguas; Em [Nguyen et al. 2012] os autores utilizam dados de infobox para agrupar esquemas de infoboxes semelhantes, a ideia é identificar um conjunto de atributos que melhor descreva classes de um determinado domínio.

Aumento de Base de Conhecimento - A tarefa de aumento de Base de Conhecimento (*Knowledge Base Augmentation*) permite a adição/melhoria de fatos em uma base de conhecimento existente. Como apresentado anteriormente, a DBpedia é a base de conhecimento referente à Wikipédia, portanto manter dados de infoboxes completos e curados contribui para o enriquecimento dessa base de conhecimento. Em [Lange et al. 2010] é proposto o iPopulator, um sistema que tem por objetivo extrair do texto de um documento da Wikipédia valores para atributos faltantes. Entidades de uma mesma classe (e.g. Ator) podem ter esquema de infobox diferentes, nesse sentido o iPopulator contribui para o enriquecimento dos infoboxes. No iPopulator os autores utilizam os dados de infoboxes para construir um esquema global para uma classe específica e para aprender padrões de valores de atributos para realizar a extração de valores no texto. Nessa mesma linha, [Moreira 2019] propõe o Deepex, a diferença entre eles é que o Deepex faz uso de algoritmos de *deep learning* para rotulagem sequencial e extração de propriedades e valores. Atualmente, na Wikipédia, existem muitas entidades sem infoboxes ou com pouca informação associada, no trabalho de [Sáez and Hogan 2018] dados do Wikidata são utilizados para criar/completar infoboxes para entidades da Wikipédia.

Sistemas de perguntas/respostas - A tarefa de perguntas/respostas consiste em colocar uma pergunta na forma de declarações de linguagem natural e um texto breve e conciso é retornado como uma resposta. Em [Morales et al. 2016] os autores utilizam dados de infobox como fonte de dados para construção de um modelo *deep learning* para esta tarefa. Em [Abbas et al. 2016] é proposto o WikiQA, um sistema de perguntas/resposta, que além de usar dados de infobox da Wikipédia também integra bases de conhecimento (e.g. DBpedia e Freebase) como fonte de dados.

Qualidade de Dados - A qualidade de dados é um requisito importante para a maioria das aplicações. Nesse sentido, trabalhos que definem métricas para avaliação e processos para melhoria na qualidade dos dados são de extrema importância. O estudo conduzido por [Wecel and Lewoniewski 2015] mostrou que a qualidade dos dados da DBpedia está relacionada com os dados fornecidos pelos Infoboxes. Os autores focaram seus estudos em definir métricas que podem ser aplicadas para avaliar a qualidade dos dados de infoboxes e como consequência disso fornecer subsidios para melhoria da qualidade da DBpedia.

Tabela 2. Comparativo entre os datasets da DBpedia, baseado em ontologia, e o conjunto de dados extraídos.

Elementos	Datasets Mappingbased	Índices Infobox
Entidades	**4.718.331**	2.796.885
Propriedades	1.366	**56.819**

5. Discussões

É esperado que a disponibilidade dos dados descritos neste artigo possa abrir diferentes oportunidades de pesquisa. Como disposto na seção anterior, dados de infoboxes são utilizados em diferentes trabalhos. Uma limitação, nos dados disponibilizados, pode ser encontrada no que se refere a formatação de valores para alguns tipos de dados (e.g. datas e propriedades multivaloradas). Como o processo de extração foi diretamente do *wikicode*, um refinamento para melhorar a apresentação desses dados pode ser realizado, inclusive essa questão é um trabalho futuro que pretende-se realizar. Com o objetivo de estabelecer uma comparação entre os dados contidos nos datasets *Mappingbased Literals* e *Mappingbased Objects*, fornecidos pela DBpedia, e nos dados extraídos por esse trabalho foram realizadas algumas análises as quais são discutidas abaixo. A Tabela 2 apresenta os dados dessa comparação para a versão em língua inglesa do dump da Wikipédia e dos datasets da DBpedia.

É possível observar que a quantidade de entidades obtidas nos datasets analisados da DBpedia são superiores a quantidade de entidades recuperadas por meio da extração realizada. Isso ocorre porque em páginas que possuem mais de um infobox o extrator da DBpedia gera uma ou mais variações dessa mesma página transformando-a em outra entidade. Uma exemplo de uma situação dessa é a da entidade *Arnold_Schwarzenegger*, que possui 2 infoboxes e no referido dataset foram criadas variações tais como: *Arnold_Schwarzenegger__1* e *Arnold_Schwarzenegger__2*. Apesar de não ser uma situação constante, algumas páginas podem ter um grande número de variações, o que justifica essa diferença nos números obtidos (e.g. *Federal_Standard_595_camouflage_colors*[17], com mais de 20 variações). Diferentemente, o parser configurado neste trabalho não realiza essas variações. Por outro lado, algumas entidades que deveriam estar presentes nos datasets da DBpedia, não estão. Isso ocorre devido à falta de mapeamento do template do infobox com a ontologia da DBpedia. Um exemplo de entidades que estão presentes nos dados coletados e não estão nos datasets *Mappingbased* são: *Ghawar_Field*, *Brent_System*, *Jamngar_Refinery*, entre outras.

Outro fator a ser destacado é a quantidade de propriedades distintas recuperadas. Como grande parte das propriedades de infoboxes não estão mapeadas para a ontologia da DBpedia, elas acabam sendo ignoradas pelo dataset analisado. Diferentemente, a extração realizada por este trabalho considera todas as propriedades, contribuindo para uma descrição mais abrangente da entidade. Isto pode ser visto quando é analisada a distribuição do tamanho dos infoboxes, que por sua vez é medido pela contagem do número de propriedades distintas. A Figura 5(a) mostra a distribuição do tamanho dos infoboxes contidos nos datasets *Mappingbased*, enquanto a Figura 5(b) mostra a distribuição do tamanho dos infoboxes dos dados coletados.

[17] `https://en.wikipedia.org/wiki/Federal_Standard_595_camouflage_colors`

Referências

Abbas, F., Malik, M. K., Rashid, M. U., and Zafar, R. (2016). Wikiqa — a question answering system on wikipedia using freebase, dbpedia and infobox. In *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, pages 185–193.

Kuzey, E. and Weikum, G. (2012). Extraction of temporal facts and events from wikipedia. In *Proceedings of the 2Nd Temporal Web Analytics Workshop*, TempWeb '12, pages 25–32, New York, NY, USA. ACM.

Lange, D., Bohm, C., and Naumann, F. (2010). Extracting structured information from wikipedia articles to populate infoboxes. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1661–1664, New York, NY, USA. ACM.

Morales, A., Premtoon, V., Avery, C., Felshin, S., and Katz, B. (2016). Learning to answer questions from Wikipedia infoboxes. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1930–1935, Austin, Texas. Association for Computational Linguistics.

Moreira, J. (2019). Extracting structured information from text to augment knowledge bases. Master's thesis, Universidade Federal de Pernambuco.

Nguyen, T., Moreira, V., Nguyen, H., Nguyen, H., and Freire, J. (2011). Multilingual schema matching for wikipedia infoboxes. *Proc. VLDB Endow.*, 5(2):133–144.

Nguyen, T. H., Nguyen, H. D., Moreira, V., and Freire, J. (2012). Clustering wikipedia infoboxes to discover their types. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2134–2138, New York, NY, USA. ACM.

Sáez, T. and Hogan, A. (2018). Automatically generating wikipedia info-boxes from wikidata. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1823–1830, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Serra, E., Cortez, E., Silva, A. S. d., and Moura, E. S. (2011). On using wikipedia to build knowledge bases for information extraction by text segmentation. *JIDM*, 2(3):259–272.

Wecel, K. and Lewoniewski, W. (2015). Modelling the quality of attributes in wikipedia infoboxes. In Abramowicz, W., editor, *Business Information Systems Workshops*, pages 308–320, Cham. Springer International Publishing.

Iudicium Textum Dataset
Uma Base de Textos Jurídicos para NLP

A. Willian Sousa[1], **Marcos Didonet Del Fabro**[1]

[1]C3SL, Centro de Computação Científica e Software Livre
Depto. de Informática – Universidade Federal do Paraná (UFPR)
CEP: 81530-900 – Curitiba – PR – Brazil

{awsousa,marcos.ddf}@inf.ufpr.br

Abstract. *The automatic text processing of natural language, with the use of probabilistic models and neural networks allows the analysis and classification of large volumes of text, leading the professionals and institutions of legal area to work more efficiently. However, the Natural Language Processing for Portuguese lacks of textual resources to support the creation and training of language models, as more deep studies related. In this article, a dataset of legal texts of the Brazilian Federal Supreme Court is presented to provide such resources. This dataset contains legal documents created by Supreme Court in its integral composition, with the subjects of the forty thousand process discriminated and, information about component sections with their respective author identified, allowing that data be used for studies of text classification, topic modeling, textual composition and others.*

Resumo. *O processamento automático de texto em linguagem natural por meio de modelos probabilísticos e redes neurais, permite a análise e classificação de grandes volumes de texto, tornando áreas como o Direito e os profissionais e instituições que o operam mais eficientes. Contudo, a área Processamento de Linguagem Natural para a Língua Portuguesa carece de recursos textuais, que permitam o treinamento de modelos robustos e o aprofundamento de estudos voltados para ela. Neste artigo, uma base de textos jurídicos obtidos da consulta pública do Supremo Tribunal Federal brasileiro é apresentada como meio de fornecer tais recursos, contando com mais de 40 mil acórdãos, além de 48 mil votos e 39 mil relatórios identificados de acordo com o seu ministro redator, provendo recursos para estudos de classificação de textos, modelagem de tópicos, composição textual e outros.*

1. Introdução

A área de Processamento de Linguagem Natural (PLN) é uma vertente da computação voltada para o estudo e criação de métodos, procedimentos e mecanimos que permitam o entendimento e processamento automático de textos escritos em linguagem natural. Essa tarefa se mostra desafiadora, pois textos em linguagem natural não possuem uma sintaxe rígida, nem tão pouco formalismos construtivos pré-determinados, exceto quando estão englobados dentro de um contexto técnico. De acordo com as tarefas de processamento a serem executadas, são necessários diferentes níveis de conhecimento linguístico, com as dificuldades de implementação proporcionais ao nível de profundidade da análise efetuada [Medeiros 1999].

Por ser a linguagem escrita, um meio utilizado pela mais diversas áreas, para registro e documentação de informações, é de alto grau a sua importância e relevância, em áreas do conhecimento como o Direito, onde o uso da linguagem técnica escrita se funde ao uso da linguagem comum na confecção de peças processuais, por meio de citações e inserções de trechos de documentos utilizados como provas ou recursos explicativos. Assim, não surpreende que o uso de PLN nesta área seja considerado um caminho natural e uma necessidade premente, considerando o grande volume de dados produzido, bem como a necessidade de manter esses dados disponíveis e consultáveis ao longo do tempo, sem causar prejuízos ao andamento processual que possui tempo e prazos definidos.

Ainda que a PLN possa contribuir significativamente para o manuseio e tratamento de informação textual, especialmente na área jurídica e que, nos últimos anos diversos métodos surgiram, facilitando o aprofundamento e aumento dos tipos e quantidades de tarefas automatizadas envolvendo texto, ainda é limitada a quantidade de recursos abertos disponibilizados livremente para a língua portuguesa. Especialmente, no tocante à dados reais que permitam treinamento de modelo probabilísticos e aqueles baseados em redes neurais. Essa escassez se dá pela dificuldade de obtenção destes dados, dos tratamentos necessários para eliminação de ruídos e a complexidade e esforço necessários para torná-los facilmente manuseáveis, sem perda de representatividade da informação original.

Com o intuito de reduzir a escassez de recursos textuais abertos para a nossa língua materna, foi construída a base apresentada neste artigo, contendo mais de 50 mil documentos jurídicos, produzidos no intervalos de 09 anos. Mais de 22 gigabytes de arquivos em formato PDF foram recuperados e processados na geração dos dados para compor a base. Para isso foram utilizados documentos que estivessem publicamente disponíveis e acessíveis, permitindo a evolução e expansão da área, fomentando a criação de modelos textuais específicos da área jurídica, tendo em vista a importância da justiça para uma sociedade democrática.

O artigo está organizado da seguinte forma. A Seção 2 descreve os documentos que compõem a base, sua finalidade, estrutura e composição. A Base de Textos Jurídicos é apresentada na Seção 3, com detalhamento de sua criação e do resultado final obtido. A Seção 4 descreve alguns desafios e limitações e, por fim, a Seção 5 exibe a conclusão do artigo.

2. Acordão

Acórdãos são documentos que resultam do julgamento por instâncias superiores do Judiciário Brasileiro, sendo um documento de estrutura rígida e bem definida. Entretanto, cada Tribunal, pela ausência de uma legislação que defina a estrutura e o conteúdo de um acórdão, ressalvado na sua independência funcional e na proposição de seu regimento interno, define a composição destes documentos, que no caso do STF, estão estruturados em seções.

A primeira seção do acórdão do STF contém uma descrição do tipo de processo a ser julgado, o nome do ministro relator do processo e as partes envolvidas. A segunda parte apresenta de modo sintético e resumido as matérias às quais o processo está relacionado, os fundamentos da decisão e uma breve descrição do próprio processo. A terceira parte, apresenta o texto do acordão propriamente dito - o resultado da votação. A próxima seção apresenta o relatório emitido pelo relator do processo, trazendo os fatos e as cir-

cunstâncias do caso julgado. A próxima seção engloba os votos dos ministros, com o primeiro voto sendo do relator e os demais ordenados de acordo com o tempo de atividade dos ministros no Tribunal, do mais novo ao mais antigo e o voto do presidente vem por último, encerrando a seção de votos.

Há ainda uma última seção, o Extrato de Ata, que repete algumas das seções como as partes, a decisão do acórdão e a indicação daqueles que estiveram presentes e ausentes ao julgamento.

3. A ITD - Iudicium Textum Dataset

Esta seção aborda a base de textos jurídicos, disponível para download no link: `http://dadosabertos.c3sl.ufpr.br/acordaos` e explicita a forma como a mesma foi concebida, detalhando os procedimentos necesários para sua criação, bem como as ferramentas utilizadas. Apresentando ainda, o resultado produzido e informações de acesso e obtenção da base.

A ITD, de acordo com a sua concepção inicial, deve englobar uma variedade de documentos que permita sua utilização em todas as esferas jurídicas e aplicação nas mais diversas tarefas aplicadas sobre distintos tipos de documentos. Para este momento de concepção da base, foram escolhidos apenas os acórdãos do Supremo Tribunal Federal (STF) publicados entre os anos de 2010 a 2018. Algumas avaliações foram feitas com documentos de anos anteriores a esse período, porém a quantidade de documentos digitalizados com baixa qualidade, fez com que optassêmos apenas por aqueles que permitissem recuperar informação de maneira precisa.

3.1. Etapas de Criação da Base

Aqui descrevemos cada uma das etapas necessárias para a criação da base, desde a captação dos dados brutos até o resultado final.

3.1.1. Recuperação dos Documentos

Os acórdãos do STF, dada a sua importância e natureza pública são disponibilizados para a sociedade através de uma página de internet que permite a consulta à jurisprudência do tribunal através do endereço `http://www.stf.jus.br/portal/jurisprudencia/`. Nesta página é possível definir filtros para a pesquisa, entre eles a data na qual foram publicados, o ministro relator, o tipo de documento e outros. Assim, a página de pesquisa gera uma requisição HTTP GET e recebe como resultado uma lista paginada de todos os documentos, cada um deles em separado, com informações básicas do processo e links para recuperar um arquivo no formato PDF contendo a integra da decisão. Esta lista, a depender do tamanho do resultado da pesquisa, torna muito lento a avaliação de cada um dos processos por um ser humano, sendo necessário um meio automático de recuperá-los.

Para recuperar todas as integras dos acórdãos de 2010 a 2018, desenvolvemos um *crawler* na linguagem Python[1], utilizando as bibliotecas *BeautifulSoup*[2], *Lxml*[3] e *Re-*

[1]https://www.python.org
[2]https://pypi.org/project/beautifulsoup4/
[3]https://lxml.de

quests[4], executando um conjunto de requisições HTTP para o serviço de consulta do STF, recebendo o resultado dessas requisições, isolando apenas as informações de interesse, executando novas requisições HTTP, recuperando os arquivos das integras e gravando-os localmente, totalizando 50.928 acórdãos. Os arquivos originais e aqueles deles derivados, foram nomeados conforme o número do acórdão a que se referem, obedecendo uma notação específica. Sobre estes documentos, utilizando a biblioteca Apache PDFBox[5] foi feita a extração de forma integral de seus textos no formatos de texto puro e Hyper Text Markup Language (HTML), sem divisão das seções componentes.

3.1.2. Separação da Seções

Inicialmente os acórdãos foram separados em partes distintas, a saber, dados do acórdão, relatório, votos e extrato da ata. Os relatórios ao serem desmembrados do documento original, produziram um único arquivo por acórdão, já os votos, produziram de um a vários arquivos, pois o voto de cada ministro foi gravado em um arquivo em separado. Já a parte referente aos dados do acórdão, gerou um único arquivo englobando as seções de Partes, Ementa e Acórdão, sendo este último a descrição do acórdão propriamente dito.

O processo de separação das seções executado sobre os arquivos PDF, foi possível graças a análise da estrutura de alguns exemplares de acórdãos por meio da qual se percebeu que cada um dos exemplares possuia um conjunto de marcadores que apontam para a página inicial de cada seção. Essa informação foi validada em todos os documentos baixados e grande parte apresentava a mesma estrutura. Exceto, documentos muito antigos - anteriores a 2010 - e aqueles compostos apenas por imagens obtidas da digitalização de documentos físicos.

A separação das seções e armazenamento das mesmas em arquivos, exigiu o desenvolvimento de um programa em Java, utilizando a biblioteca Apache PDFBox, para ler cada uma das seções e gerar as chamadas da própria biblioteca, necessárias para extração dos textos em formato HTML e seu consequente armazenamento. Cada arquivo gerado foi nomeado de forma que permita identificar o acórdão ao qual pertence, a seção a que se refere, o intervalo de páginas correspondentes no documento original e a sua posição relativa às demais seções. Para seções como Votos e Relatório, é incluído o nome do redator do texto daquela seção. Por exemplo, o arquivo `1_Voto_MIN_ROSA_WEBER_pg_6_13_seq_3.html` refere-se à um dos votos do acórdão número 01, emitido pela Ministra Rosa Weber e está localizado entre as páginas 06 e 13 no documento original, sendo a 3ª parte do documento. Mais formalmente, a nomenclatura dos arquivos obedece a seguinte notação:

```
<NUM_ACORDAO>_<SECAO>_pg_<PGINI>_<PGFIM>_seq_<SEQ>.<EXT>
<NUM_ACORDAO>::=<NUMERO>
<SECAO>::=<IDSECAO>|<ID_MINISTRO>
<IDSECAO>::=Ementa_e_Acordao|Relatorio_<ETIQUETA_MINISTRO>|
            Voto_<ETIQUETA_MINISTRO>
<ETIQUETA_MINISTRO>::=<TIPOID>|<TIPOID>_<MINISTRO>
<TIPOID>::=MINISTRO_PRESIDENTE|VICE_PRESIDENTE|MIN
<MINISTRO>::=GILMAR_MENDES|MARCO_AURELIO|CARLOS_VELLOSO|
```

[4]https://2.python-requests.org
[5]https://pdfbox.apache.org

```
                CEZAR_PELUSO|ELLEN_GRACIE|ALEXANDRE_DE_MORAES|
                EDSON_FACHIN|ROSA_WEBER|CARMEN_LUCIA|LUIZ_FUX|
                TEORI_ZAVASCKI|DIAS_TOFFOLI|SEPULVEDA_PERTENCE|
                CELSO_DE_MELLO|ROBERTO_BARROSO|JOAQUIM_BARBOSA|
                AYRES_BRITTO|EROS_GRAU|RICARDO_LEWANDOWSKI
<PGINI>::=<PG>
<PGFIM>::=<PG>
<PG>::=<NUMERO>
<SEQ>::=<NUMERO>
EXTENSAO::=HTML|TXT
<NUMERO>::=<NUMERO><DIGITO>|<DIGITO>
<DIGITO>::=0|1|...|9
```

Essa nomenclatura permite a obtenção da integra do acordão por meio de suas partes, além de possibilitar a utilização das seções com identificação do redator em processos de aprendizagem, comparação, agrupamento e outras.

3.1.3. Pré-processamento de Texto

Concluído o processo de separação das seções dos acórdãos, seguiu-se uma limpeza de textos desnecessários, como dados de assinatura digital dos documentos, numeração de páginas e outros inseridos no cabeçalho e rodapé dos textos. Apesar dos dados terem sido extraídos em dois formatos distintos, o HTML foi escolhido para esta etapa, dada a facilidade de identificação e separação das estruturas sintáticas do texto por meio de suas tags. A figura 1 ilustra as seções da primeira parte do documento. Há ainda, outros textos como assinaturas, locais e datas, indicação final e inicial do relator, nome do ministro no voto e outros que foram avaliados e eliminados dos documentos por não serem considerados relevantes para os objetivos de construção da base. Entretanto, essas informações ainda podem ser recuperadas na própria base, pois mantivemos uma cópia do texto integral para cada documento.

Para cada elemento da seção das partes do processo, foi necessária uma separação entre a indicação de seu tipo e o nome da parte. Da ementa e do texto do acórdão, foram retirados apenas os textos identificadores da seção e dados como assinaturas e datas. A separação desses textos foi feita por meio de expressões regulares e análise da estrutura de marcação dos documentos HTML. Deve-se ressaltar que a maior parte dos procedimentos de limpeza e organização dos textos foi feita baseando-se no conteúdo da extração em HTML e a mesma não pode ser aplicada diretamente sobre os arquivos em texto puro.

Mesmo o acórdão sendo uma síntese do julgamento e da decisão, em algumas situações pode apresentar uma quantidade de páginas expressiva, com a seção de Votos ocupando a maior parte, por conta das justificativas dos ministros e debates que podem ocorrer em plenário e que são registrados no documento. Estas informações e outras que figurem na seção de votos, são extraídas e processadas, porém no arquivo gerado, os mesmos não tem uma identificação do redator, pelo fato daquele texto ser um registro de um debate técnico e não pertencer a um único ministro. Assim, essa informação se encontra disponível na ITD, porém sem identificação, sendo possível consultá-la em separado para outras atividades de pesquisa.

A seção de Extrato da Ata, documentos extras, textos documentais e os debates

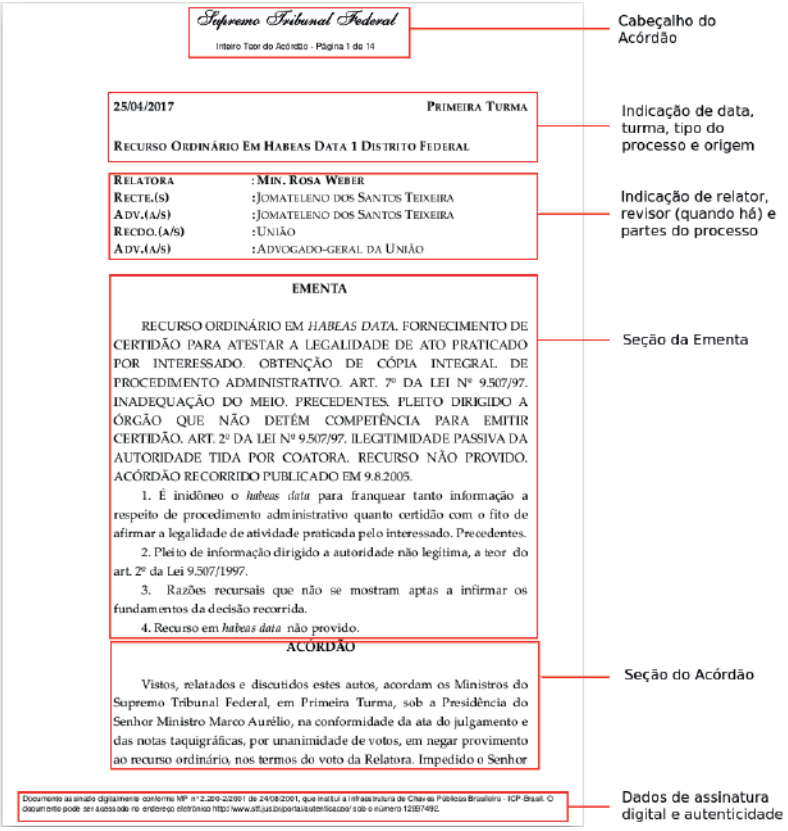


Figura 1. Partes do documento do acórdão

não sofreram nenhum tipo de pré-processamento, exceto remoção de cabeçalho e dados de autenticidade do documento.

3.1.4. Base de Documentos NoSQL

O JavaScript Object Notation (JSON)[6] é uma notação que permite compartilhamento e publicação de dados em diversas áreas e para os mais diversos propósitos. Sendo comum a sua utilização para interação remota entre aplicações de Internet, intercâmbio de dados científicos e para disponibilização de dados públicos abertos [Baazizi et al. 2019]. O formato JSON foi escolhido para armazenamento e estruturação da nossa base de textos jurídicos, por conta da sua flexibilidade, interoperabilidade, difusão e uso tanto em aplicações da indústria, quanto acadêmicas. A estrutura de documentos neste formato, permite expansão futura sem perda de compatibilidade com versões anteriores dos dados.

Para facilitar o armazenamento e manipulação, os dados em formato JSON foram inseridos em uma base de documentos MongoDB[7], um sistema gerenciador de documentos NoSQL que permite utilização sem o uso de um esquema rígido de descrição da estrutura dos seus documentos. Neste sistema os dados são armazenados em um formato chamado BSON (Binary JSON), uma representação binária da serialização de documentos JSON. A inserção dos dados textuais na base NoSQL, após as etapas de pré-

[6]https://www.json.org
[7]https://www.mongodb.com

processamento, foi feita por meio de um programa desenvolvido na linguagem Python, estando o mesmo disponível juntamente com a base.

Além de armazenar os documentos de forma acessível, também é necessário manter a representação do documento do acórdão em JSON o mais fiel possível ao seu original. Com esse intuito, a estrutura do documento foi concebida da seguinte forma:

```
{
  _id : <Identificação interna do MongoDB>,
  caminho : <Caminho do diretorio da base de texto>,
  arquivohtml : <Caminho do arquivo da integra em HTML>,
  arquivopdf : <Caminho do arquivo da integra em PDF>,
  arquivotxt : <Caminho do arquivo da integra em TXT>,
  arquivossecoes : {
    ATA : <Caminho do arquivo do Extrato da Ata>,
    EMENTA_ACORDAO : <Caminho da seção Ementa e Acórdão>,
    OUTROS : [ <Caminho dos outros arquivos do Acórdão> ],
    RELATORIO : <Caminho do arquivo do Relatório>,
    VOTOS : {
        <Nome arquivo do voto> : <Caminho do arquivo do Voto>
    }
  },
  numero : <Número do Acórdão>,
  partes : [
     {
       nome : <Nome da parte>,
       sigla : <Sigla da parte>,
       tipo : <Identificação de tipo da parte>
     }
  ],
  ementa : { texto : <Texto da Ementa>   },
  acordao : { texto : <Texto do acórdão> },
  extratoata : { texto : <Texto do Extrato da Ata> },
  formato : <Formato escolhido para pré-processamento>,
  integrahtml : <Integra em formato HTML>,
  integratxt : [  <Integra em TXT separada por linhas> ],
  relatorio : {
        relator : <Ministro relator>,
        texto : <Texto do relatório>
  },
  votos : [
    {
          texto : <Texto do voto>,
          votante : <Ministro votante>
        }
  ]
}
```

Após os devidos tratamentos dos dados textuais, a base de documentos conta com 41353 documentos, aproximadamente 70% da quantidade de arquivos recuperados inicialmente. Essa diferença se deve ao fato de alguns desses documentos, apesar de estarem em formato PDF, não permitirem a extração de seus textos sem um prévio processamento de reconhecimento ótico de caracteres. Como nosso interesse era a obtenção de textos de forma precisa, esses documentos foram ignorados durante os processos de conversão.

3.2. Estatísticas da Base

Apresentamos aqui, algumas informações sobre os quantitivos da ITD considerando apenas as informações armazenadas no MongoDB.

A base conta com um total de 41.353 documentos, com igual quantidade de relatórios e 56.250 votos. A maioria dos acórdãos possui apenas um texto de voto, como mostra a figura 2, na qual são apresentados dados dos acórdãos pela sua quantidade de votos. Já nas figuras 3 e 4 são apresentados, respectivamente os quantitativos de votos e relatórios dos últimos 12 ministros em atividade no STF. Essas quantidades não cobrem todos os votos da base, nem tão pouco os relatórios.

As quantidades de votos e relatórios emitidos pelos ministros dependem das funções que estes acumulam dentro do tribunal e as incumbências que lhes são determinadas pelo regimento interno do órgão. Assim, quando na condição de presidente de turma ou do próprio STF, um ministro poderá ter processos que são exclusivos de sua competência. Os votos e relatórios emitidos sob essas condições estão identificados na base com a etiqueta `MINISTRO_PRESIDENTE`. A identificação dos redatores destes votos e relatórios não foi executada, pois os mesmos totalizam apenas 1180 votos e relatórios, o que representa apenas 2,85% do total de relatórios e 2,09% do total de votos. Entretanto, caso os mesmos sejam necessários para alguma atividade, há como identificá-los com um trabalho de extração de informação textual, pois no corpo do texto do voto e do relatório há uma indicação do nome do ministro redator.

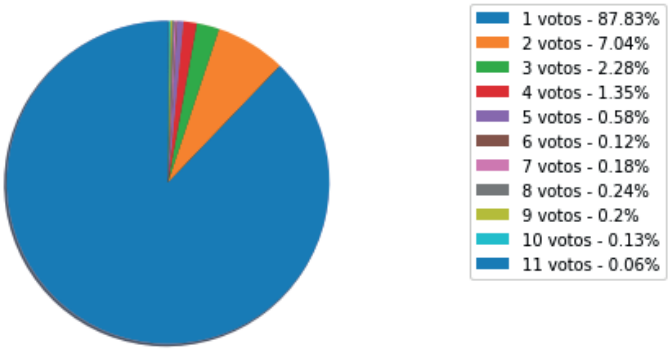


Figura 2. Acórdãos por Número de Votos

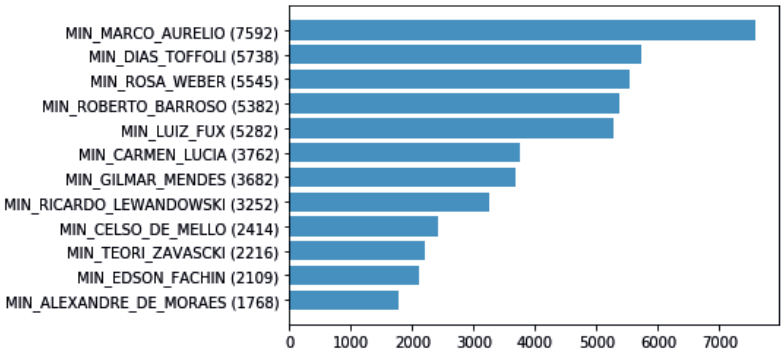


Figura 3. Votos por Ministro

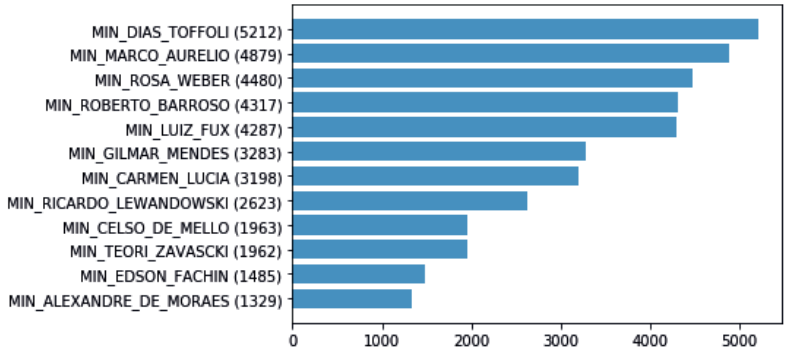


Figura 4. Relatórios por Ministro

3.3. Disponibilização da Base

A ITD pode ser obtida através do endereço `http://dadosabertos.c3sl.ufpr.br/acordaos`, onde estão disponibilizados os arquivos originais das integras e das seções nos formatos PDF, HTML e TXT. Também estão disponíveis os dados em formato JSON e os mesmos podem ser utilizados para importação em qualquer gerenciador de documentos que ofereça suporte ao formato, ainda que os mesmos tenham sido exportados do MongoDB, versão 4.0.9, a qual se recomenda o uso por questões de facilitação do uso dos programas desenvolvidos.

Os arquivos estão disponibilizados conforme a estrutura de diretórios detalhada na tabela 1, com a ressalva que no diretório *BaseITD/ITD* do site há apenas um arquivo compactado que comporta a estrutura aqui descrita. Já no diretório *BaseITD/tools* estão todos os programas desenvolvidos para criação da base, além de exemplos de como utilizá-los e no diretório *BaseITD/json* estão os arquivos prontos para importação em bases NoSQL.

Estão disponíveis 03 arquivos em formato JSON, onde o primeiro contém toda a base de acórdãos, o segundo apenas os votos e o terceiro apenas os relatórios. As informações destes dois últimos arquivos podem ser obtidas a partir da base completa, tendo sido separados apenas para facilitar o uso em tarefas de PLN, como treinamento de modelos de representação vetorial de palavras e classificação de textos, nos quais estamos trabalhando e que serão futuramente disponibilizados de forma aberta.

Diretório	**Conteúdo**
`BaseITD/ITD`	Diretórios nomeados de acordo com os números dos acórdãos
`BaseITD/ITD/<NNN>/html`	Arquivos html extraídos do acórdão
`BaseITD/ITD/<NNN>/txt`	Arquivos txt extraídos do acórdão
`BaseITD/tools`	Scripts e programas para extração e geração da base, além de notebooks de uso da base
`BaseITD/json`	Arquivos json contendo documentos da base

Tabela 1. Estrutura de diretórios da ITD

4. Desafios e Limitações

Os principais desafios encontrados na criação da base estão relacionados à falta de padronização e aos meios e formatos como os documentos se encontram disponíveis, exigindo o desenvolvimento de ferramentas específicas e obrigando a uma validação e análise contínuas dos dados obtidos, como meio de assegurar a sua qualidade.

Uma vez que formatos como PDF não são pensados para estruturação de texto, mas sim para a sua apresentação, analisar a estrutura do texto formatado e definir uma maneira de, a partir dos marcadores de formatação, obter a estrutura desejada e permitir que a mesma pudesse ser reconstruída tomando-se como base os dados já extraídos e processados, foi certamente a barreira mais importante a ser transposta durante o desenvolvimento da base.

A definição de métodos e estruturas de dados que permitissem extrair os dados, possibilitando a reconstrução da estrutura inicial do documento, foi feita por meio da geração de um formato intermediário, no caso o HTML, que apesar de também ser uma linguagem de formatação, facilitou o processo de extração. Assim, a conversão de PDF para HTML foi responsável pela manutenção da macro-estrutura do documento e a extração a partir do HTML, pela geração dos dados componentes da base. Esta última parte, exigiu um trabalho de garantia da qualidade dos resultados através da análise contínua e, por vezes, obrigou a reorganização de toda a cadeia de procedimentos de conversão e extração dos documentos para garantir a qualidade desejada.

No tocante às limitações da base, o fato da mesma conter apenas um único tipo de documento - acordãos emitidos pelo STF - e não permitir a sua utilização para análise de acórdãos de outros tribunais sem adaptações, pode reduzir a velocidade da sua expansão. Entretanto, como a mesma foi pensada e estruturada para não aderir a uma definição rígida dos dados, poderá abarcar uma grande variedade de tipos de documentos e comportar novas informações. Aumentando as suas possibilidades de uso, que, atualmente, já não são limitadas apenas às áreas tecnológicas, pois o seu conjunto de textos pode subsidiar, por exemplo, pesquisas nas áreas de estudos jurídicos e da linguagem escrita e falada, cobrindo um espectro de estudo que vai desde a forma como os ministros escrevem e estruturam os seus votos e relatórios, até as motivações e valorações utilizadas para embasar as suas decisões.

5. Conclusão

Neste artigo apresentamos a *Iudicium Textum Dataset*, uma base de textos jurídicos em Língua Portuguesa composta por documentos dos acórdãos do Supremo Tribunal Federal, que até onde sabemos se trata da primeira base deste tipo, tratada e disponibilizada abertamente. Na literatura relacionada é possível encontrar trabalhos como [Braz et al. 2018], [Da Silva et al. 2018] e [de Araujo et al. 2018] nos quais a criação de bases de textos jurídicos é citada, porém apenas a última está disponível publicamente e é voltada especificamente para a tarefa de reconhecimento de entidades nomeadas, contendo uma pequena quantidade de documentos, ainda que os mesmos apresentem considerável variabilidade.

Esperamos que a publicação dos dados consolidados e em formato aberto auxilie na ampliação da área de PLN, fomente o desenvolvimento de novas aplicações, a criação de novas bases e a melhoria e expansão da própria ITD, visando uma justiça mais eficiente e, cada vez mais, acessível a todos.

Referências

Baazizi, M.-A., Colazzo, D., Ghelli, G., and Sartiani, C. (2019). Schemas and types for json data: From theory to practice. In *Proceedings of the 2019 International Conference on Management of Data*, pages 2060–2063. ACM.

Braz, F. A., da Silva, N. C., de Campos, T. E., Chaves, F. B. S., Ferreira, M. H. S., Inazawa, P. H., Coelho, V. H. D., Sukiennik, B. P., de Almeida, A. P. G. S., de Barros Vidal, F., Bezerra, D. A., Gusmao, D. B., Ziegler, G. G., Fernandes, R. V. C., Zumblick, R., and Peixoto, F. H. (2018). Document classification using a bi-lstm to unclog brazil's supreme court. *CoRR*, abs/1811.11569.

Da Silva, N. C., Braz, F., de Campos, T., Gusmao, D., Chaves, F., Mendes, D., Bezerra, D., Ziegler, G., Horinouchi, L., Ferreira, M., et al. (2018). Document type classification for brazil's supreme court using a convolutional neural network. In *The tenth international conference on forensic computer science and cyber law-ICoFCS*, pages 7–11.

de Araujo, P. H. L., de Campos, T. E., de Oliveira, R. R., Stauffer, M., Couto, S., and Bermejo, P. (2018). Lener-br: A dataset for named entity recognition in brazilian legal text. In *International Conference on Computational Processing of the Portuguese Language*, pages 313–323. Springer.

Medeiros, M. B. B. (1999). Tratamento automático de ambigüidades na recuperação da informação.

JusBD: Um Banco de Dados para Obtenção de Informações do Poder Judiciário

Weverton Ryan Ribeiro da Mata[1], **Danilo B. Seufitelli**[2], **Michele A. Brandão**[1]

[1]Instituto Federal de Minas Gerais (IFMG) – Campus Ribeirão das Neves
Ribeirão das Neves – MG – Brasil

[2]Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

ryanmata40@outlook.com,daniloboechat@dcc.ufmg.br,
michele.brandao@ifmg.edu.br

Abstract. *Digital forensics is a science that seeks to identify, preserve, retrieve, analyze, and present evidence of crime in data. In Brazil, the Law of Information Access allows public data to be audited or undergo a forensic analysis. However, many of these data are in PDF or in large spreadsheets, which makes any type of analysis difficult. Thus, this work presents JusBD, a relational database that represents an advance for the analysis of public data by integrating datasets of the Brazilian judiciary. Therefore, JusBD allows obtaining various information about the judiciary and enables a forensic analysis.*

Resumo. *A forense digital é uma ciência que visa a identificação, preservação, recuperação, análise e apresentação de evidências de crimes em dados. No Brasil, a Lei de Acesso à Informação possibilita que dados públicos sejam auditados ou passem por uma análise forense. Porém, muitos desses dados estão em PDF ou em grandes planilhas, o que dificulta qualquer tipo de análise. Dessa forma, este trabalho apresenta JusBD, um banco de dados relacional que representa um avanço para a análise de dados públicos por integrar bases de dados do poder judiciário brasileiro. Assim, o JusBD permite a obtenção de diversas informações sobre o poder judiciário e possibilita uma análise forense.*

1. Introdução

Nas últimas décadas, um tipo de crime que tem se difundido é o ocorrido em meios eletrônicos ou digitais. Identificação não autorizada de dados de saúde de pacientes [Anderson and Williams 2018], ocorrência de fraudes bancárias [Becker et al. 2017] e roubos de identidade [Dadkhah et al. 2018] são exemplos de tais crimes. O grande volume de informações ao qual a internet expõe os indivíduos todos os dias e a popularização dos dispositivos digitais são algumas das causas para essa difusão. Se por um lado esses dois fatores contribuem para o aumento do número de vítimas de crimes, por outro lado, também aumentam a quantidade de conteúdo, atividades e rastros, que podem servir de evidências deixadas por ofensores.

Os dados sobre tais evidências podem ser identificados, preservados, recuperados, analisados e apresentados por uma ciência denominada *Forense Digital* [Guarino 2013].

Com frequência, essas evidências são relacionadas a crimes virtuais. Além disso, a forense digital auxilia a apoiar ou refutar algum tipo de suposição, reconstruir eventos criminais e predizer ações não autorizadas [Guarino 2013]. Por exemplo, Greengard [2012] descreve o caso de um funcionário do Burger King em Ohio (estado dos Estados Unidos) que postou fotos em uma rede social com os pés em bandejas de alfaces. Diferentes usuários da rede social fizeram uma simples investigação forense e descobriram a localização do Burger King. Como resultado, três funcionários foram demitidos.

Nesse contexto, a Lei nº 12.527, também conhecida como "Lei de Acesso à Informação", foi sancionada em 18 de novembro de 2011 com o objetivo de promover a Transparência da Administração Pública Brasileira. Tal lei foi modificada pelo Decreto nº 9.690, de 23 de janeiro de 2019, que aumentou a quantidade de pessoas aptas a decidirem sobre o sigilo de dados públicos. Apesar dessa mudança, ainda há diversas bases de dados públicos disponíveis na Web. Entretanto, muitos desses dados estão em PDF (Portable Document Format) ou em uma grande planilha, muitas vezes, com várias abas. Isso dificulta uma auditoria nesses dados e até mesmo uma análise forense.

Assim, este trabalho representa um avanço para a análise de dados públicos ao baixar, processar, unir diferentes planilhas e modelar um banco de dados relacional para bases de dados públicas do Poder Judiciário brasileiro disponibilizadas no site do Conselho Nacional de Justiça (CNJ)[1]. Também são adicionadas informações de relacionamentos sociais entre profissionais do judiciário, por meio da criação da estrutura de uma rede social, que podem auxiliar em uma análise forense. Ademais, tais dados podem ser enriquecidos posteriormente com informações de redes e mídias sociais.

Nossas contribuições são assim resumidas. Primeiro, são discutidos os trabalhos relacionados com ênfase nos dados utilizados (Seção 2). Segundo, são apresentados o JusBD, um banco de dados relacional que permite a fácil extração de informações do Poder Judiciário brasileiro, até mesmo uma análise forense, e a metodologia para construção do JusBD (Seção 3). Terceiro, são discutidas aplicações reais do JusBD (Seção 4). Finalmente, são descritos os desafios e limitações deste trabalho (Seção 5), bem como as principais conclusões (Seção 6).

2. Trabalhos Relacionados

A área de forense digital é recente e ainda tem espaço para diferentes estudos. Ao buscar pelo termo *forensic* na DBLP[2], apenas 6.492 publicações são retornadas no período de 1988 até 2019, apenas a partir do ano 2000 que são retornadas mais de dez publicações. Entretanto, a quantidade de dados coletados para realização de pesquisas em forense tem aumentado recentemente [Guan et al. 2019], alguns deles são brevemente descritos aqui.

2.1. Conjuntos de Dados para Pesquisa Forense

Nesta seção, são descritos os conjuntos de dados frequentemente utilizados em pesquisas forense por permitirem a transparência e exposição de informações. Note que nenhum deles são dados públicos brasileiros.

[1]CNJ - uma instituição pública cujo objetivo é melhorar o trabalho do sistema judiciário brasileiro, principalmente, em relação ao controle e à transparência administrativa e processual: `http://www.cnj.jus.br/transparencia`

[2]DBLP: `https://dblp.uni-trier.de/search?q=forensic`

GovDocs. O corpus do Govdocs é uma grande coleção de aproximadamente 1 milhão de documentos governamentais que estão disponíveis gratuitamente para pesquisa, fornecidos pelo site da Digital Corpora[3].

T5 File Corpus. T5-corpus é um subconjunto do Govdocs que foi criado por Roussev [2011] e contém 4.457 arquivos de vários tipos e é comumente usado para testar a correspondência aproximada (*bitwise*), por exemplo, por Breitinger e Roussev [2014].

***BOSSBase* e *BOSSRank*.** Conjunto de dados com uma grande quantidade de imagens reais, gráficos gerados por computador e imagens forjadas contaminadas com esteganografia[4]. Alguns desses conjuntos de dados vêm de sites como o *Break the Steganography System (BOSS)*, que hospeda um desafio com um banco de dados de teste de imagens em escala de cinza de 1.000 512 512 PGM e um banco de dados de treinamento de 9.074 imagens da capa.

Corpus msx-13. Roussev e Quates (2013) criaram o corpus msx-13 que contém 22.000 arquivos aleatórios gerados por usuários do MS Office 2007 (por exemplo, docx, xlsx, pptx) rastreados da Internet. Tal conjunto de dados pode ser utilizado, por exemplo, para teste de classificadores que podem ser utilizados para identificar atividades criminosas.

Emails/Enron. Esse conjunto de dados foi coletado e preparado pelo Projeto CALO (Um Assistente Cognitivo que Aprende e Organiza). Ele contém dados de cerca de 150 usuários, a maioria da gerência sênior da Enron, organizados em pastas. O corpus contém um total de cerca de 0,5 milhões de mensagens.

M57-patents Scenario. Esse cenário inclui quase um terabyte de informações com 50 imagens de disco, despejos de memória e pacotes de rede. Oferece uma variedade de amostras de conjuntos de dados gerados por experimentos (por exemplo, dados de RAM, emails, imagens de unidades de disco, etc.). Isso é continuado e usado pelo Projeto CFReDS, BOSS e Imagens de Teste da Ferramenta *Digital Forense* (DFTT).

Real Drive Corpus. O Real Data Corpus (RDC) é uma coleção de dados brutos extraídos de dispositivos de transporte de dados que foram comprados no mercado secundário em todo o mundo. Muitos estudos demonstraram que discos rígidos, telefones celulares, cartões de memória USB e outros dispositivos portadores de dados são frequentemente descartados por seus usuários originais sem que os dados sejam primeiramente limpos ou eliminados. Ao comprar esses dispositivos e extrair seus dados, criaram um conjunto de dados que simula os dados da forma como são encontrados no mundo real.

2.2. Estudos Forense em Redes Sociais

Além de criar o JusBD com dados do Poder Judiciário brasileiro disponíveis no CNJ, este trabalho também modela uma rede social entre os profissionais do judiciário. A ideia é que tal rede seja posteriormente incrementada com informações de outras fontes, como mídias sociais. Assim, também exploramos aqui como as redes sociais são utilizadas em estudos forenses.

[3]Digital Corpora: `https://digitalcorpora.org/`

[4]Esteganografia: A arte de esconder informações de forma a torná-las ocultas [Artz 2001].

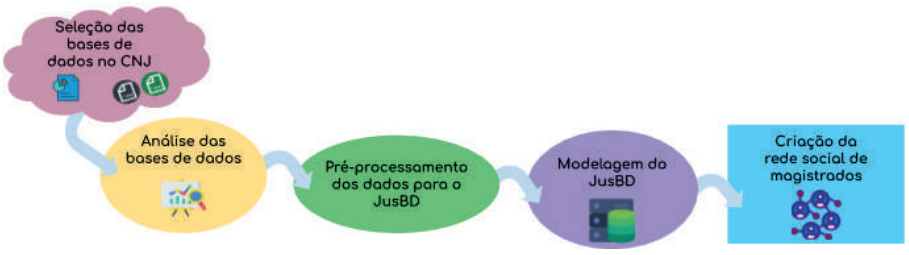


Figura 1. Cinco principais etapas da metodologia para construção do JusBD.

Nesse contexto, Al Mutawa *et al.* [2012] realizam análises forenses de três redes sociais (Facebook, Twitter e MySpace) em smartphones com o objetivo de determinar se as atividades realizadas por meio desses aplicativos são armazenadas na memória interna do dispositivo. Nesse caso, a quantidade, a importância e os locais dos dados que podem ser encontrados e recuperados da imagem lógica de cada dispositivo foram determinados. Além disso, Arshad *et al.* [2019] apresentam uma revisão do estado da arte na perícia da mídia social. Já Walnycky *et al.* [2015] descrevem um estudo forense experimental em vinte aplicativos de mensagens do sistema operacional Android. Para ilustrar o potencial de aquisição de evidências digitais a partir do dispositivo móvel, dados em trânsito e dados armazenados em servidores. Por fim, Zainudin *et al.* [2010] apresentam um processo de investigação forense em redes sociais, com o intuito de propor um modelo padrão para a investigação forense em tais redes.

Note que esses trabalhos utilizam dados de interações online entre usuários e não focam em dados da administração pública. Assim, este trabalho se diferencia dos demais não apenas por considerar dados do Poder Judiciário brasileiro, mas também por elaborar uma estrutura de rede social entre profissionais do judiciário que permite, por exemplo, uma análise forense dos salários[5].

3. Metodologia

Esta seção descreve as cinco principais etapas para criação do JusBD apresentadas na Figura 1, conforme segue.

Seleção das bases de dados no CNJ. Para a construção do JusBD[6], buscas e análises na plataforma do CNJ foram realizadas. Note que o portal do CNJ permite acesso a diferentes dados do Poder Judiciário brasileiro, tais como execução orçamentária e financeira do CNJ, e despesas com pessoal, licitações e contratos. Após a análise desse portal, seis bases de dados foram selecionadas e baixadas no formato CSV (Comma-separated values), XLS (Microsoft Excel file format) ou como tabelas para a montagem do banco de dados. Também foram encontrados relatórios em formato PDF, os quais não foram considerados pela dificuldade de extração dos dados e falta de padronização no PDF. As seis bases de dados foram Relatório do censo do Poder Judiciário, Justiça em números (dados no Qlikview), CNJ em número (dados no Qlikview), Remuneração dos magistrados, Estatísticas

[5]Supersalários de juízes é uma forma de corrupção: `http://www.sinjus.org.br/farra-dos-supersalarios-de-juizes-e-uma-forma-de-corrupcao/` e `https://www.em.com.br/app/noticia/politica/2019/01/22/interna_politica,1023348/juizes-e-desembargadores-de-mg-receberam-mais-de-r-100-mil.shtml`

[6]JusBD: `https://github.com/miabrandao/JusBD`

Tabela 1. Dados do Poder Judiciário brasileiro disponibilizado no site do CNJ.

Dado	Arquivos Disponíveis	Tamanho Total	Origem	Data Criação/ Modificação	Link Acesso
Remuneração dos Magistrados	1.523 arquivos .xls gerando 7.165 .csv	0,99 GB somente de arquivos .xls	Tribunais brasileiros	Nov/2017 até Mar/2019	https://bit.ly/2vMycsi
Estatísticas oficiais do Poder Judiciário	2 arquivos .csv	>6,36MB	Tribunais brasileiros	2004-2017	https://bit.ly/2VKxp2w
1º, 2º e 3º Balanço Socioambiental do Poder Judiciário	3 arquivos .xls gerando 3 .csv	13,3 MB	Tribunais de justiça/Seções jud./TREs/CNJ	2015–2017	https://bit.ly/2JDMhxZ

oficiais do Poder Judiciário, e 1º, 2º e 3º Balanço socioambiental do Poder Judiciário.

Análise das bases de dados. Após a seleção das seis bases de dados, primeiramente, foi analisado ao que elas se referiam, finalizando com a exclusão de três bases para a criação do conjunto. Das três bases de dados excluídas, duas foram desconsideradas por estarem em uma plataforma denominada Qlikview[7]. Notem que foram realizadas algumas tentativas, mas não foi possível obter dados dessa plataforma de forma organizada e/ou sucinta. Já a terceira base de dados foi considerada não adequada por ser utilizada em um relatório do censo do Poder Judiciário. Tal relatório possui apenas uma tabela em formato XLS com a quantidade e porcentagem de respostas por tribunal. Ademais, os dados não eliminados por essas análises são referentes à remuneração dos magistrados, estatísticas feitas anualmente de 2004 até 2017, e a base de dados do primeiro e segundo balanço socioambiental do Poder Judiciário, conforme mostra a Tabela 1. Tais bases foram selecionadas por conterem dados relevantes e com padrão para serem extraídos de forma automatizada. Note a quantidade extensa de arquivos que foram considerados e unidos para compor o JusBD. Após a análise das bases, é necessário realizar um pré-processamento dos dados.

Pré-processamento dos dados para o JusBD. Essa etapa consiste na limpeza e tratamento dos dados antes de serem inseridos no JusBD. Ao analisar os dados, constatou-se que os valores de cada atributo estavam entre aspas simples o que transformava todos em VARCHAR (Variable Character Field, um conjunto de dados de caractere de tamanho não definido) ao automaticamente inserir no banco de dados. Para resolver isso, realizamos a substituição das aspas simples por nenhum caractere apenas nos campos de valores numéricos. Ademais, também verificou-se que para colunas que o valor deveria ser numérico, haviam alguns caracteres com R$ que impediam a inserção no JusBD como um tipo int ou float, então esses caracteres também foram removidos. Após esses tratamentos, uma segunda etapa foi executada que consiste na alteração do formato dos dados para a importação no JusBD. Tal alteração foi feita por meio da utilização de um software de conversão chamado XLS to CSV converter. Assim, 1.523 arquivos .xls com cinco tabelas (abas) cada foram convertidos em 7.615 arquivos .csv referentes à base de dados de remuneração dos magistrados. Essa quantidade de arquivos .csv foram gerados, pois eles representam cada aba do arquivo .xls. Também foram considerados 2 arquivos .csv de estatísticas oficiais do Poder Judiciário, e por fim, 3 arquivos .xls com apenas uma aba que gerou 3 arquivos .csv referentes ao primeiro, segundo e terceiro balanço socioambiental do Poder Judiciário.

[7]Qlikview: `https://www.qlik.com`

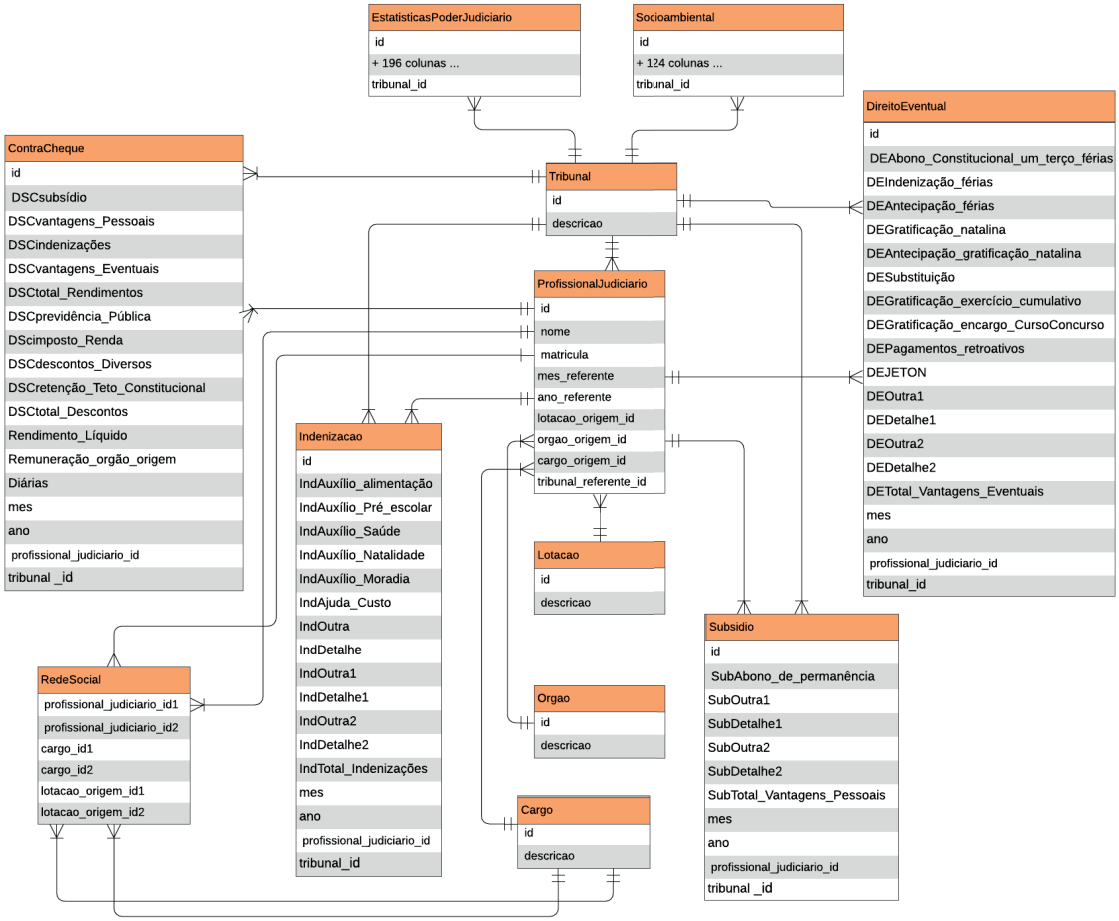


Figura 2. Esquema lógico do JusBD.

Construção do JusBD. Para a construção do banco de dados, utilizou-se um Sistema de Gerenciamento de Bancos de Dados (SGBD) relacional denominado MySQL Workbench[8] e o HeidiSQL[9], um programa livre e cliente de código-aberto para o MySQL que possibilita a consulta e a importação dos arquivos CSV para as tabelas. Dessa forma, foi elaborado um esquema lógico, conforme mostra a Figura 2, para unir as três bases de dados em um banco de dados de forma a facilitar a consulta e o acesso aos dados. Note que são doze tabelas, dez delas são resultantes das três bases de dados baixadas do CNJ e uma tabela chamada de RedeSocial foi construída a parte pelo processamento dos dados das tabelas ProfissionalJudiciario e ContraCheque. Finalmente, a Tabela 2 mostra uma breve descrição dos dados armazenados por cada tabela, a quantidade de linhas e colunas em cada tabela do JusBD[10]. Note que a tabela EstatisticasPoderJudiciario é a que possui maior quantidade de colunas[11], as quais representam, de maneira geral, a realidade dos tribunais brasileiros em números para subsidiar a Gestão Judiciária brasileira, feita anu-

[8]MySQL Workbench: `https://www.mysql.com/products/workbench/`

[9]HeidiSQL: `https://www.heidisql.com/`

[10]Exemplos de consultas SQL no JusBD: `https://github.com/miabrandao/JusBD/wiki/Queries`

[11]Em `https://github.com/miabrandao/JusBD/blob/master/Variveis.zip`, é possível consultar a definição das colunas. O mesmo para a tabela Socioambiental.

Tabela 2. Quantidade de linhas e colunas em cada tabela no JusBD.

Tabela	Armazena	# linhas	# colunas
ProfissionalJudiciario	Dados de profissionais do Poder Judiciário, tais como juizes, desembargadores, magistrados inativos, etc	288.341	9
Tribunal	Nome de sete tipos de tribunais brasileiros	7	2
Indenizacao	Dados sobre as indenizações recebidas pelos profissionais do Poder Judiciário	265.230	18
Subsídio	Dados sobre os subsídios recebidos por cada profissional do judiciário	279.289	11
ContraCheque	Dados sobre o pagamento do salário dos profissionais do judiciário	348.532	18
DireitoEventual	Detalhes sobre direitos trabalhistas dos profissionais do judiciário	258.409	20
Lotacao	Descrição do local da comarca	15.463	2
Orgao	Descrição dos órgãos judiciários brasileiros onde cada profissional trabalha	365	2
Cargo	Nome do cargo ocupado por cada profissional	401	2
RedeSocial	Dados sobre pares de profissionais do judiciário que trabalham em um mesmo órgão	501.009.904	6
EstatisticasPoderJudiciario	Estatística do Poder Judiciário que permitem comparações, diagnósticos, análises estatísticas, mensurações e avaliações de desempenho ou produtividade de órgãos, unidades, magistrados e servidores, para subsidiar a tomada de decisões	308	198
Socioambiental	Dados que auxiliam na realização de estudos e apresentação de propostas de integração das metas do Poder Judiciário com as metas e indicadores dos Objetivos de Desenvolvimento Sustentável	10.293	126

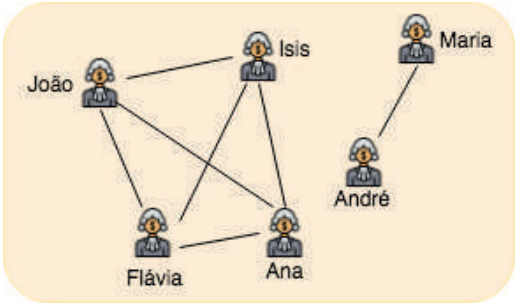


Figura 3. Exemplo de rede social de profissionais que trabalham em um mesmo órgão do Poder Judiciário.

almente desde 2004. Exemplos de colunas são *pe* que descreve as despesas com pessoal e encargos, *cpj* que armazena o número total de casos pendentes, e *G7* que representa a despesa total por habitante. A tabela Socioambiental também possui muitas colunas, 126 ao todo, que representam, em geral, ações de sustentabilidade empreendidas no Poder Judiciário brasileiro. Exemplos dessas colunas são *gtm* que armazena os gastos com telefonia móvel, *gct* que descreve o gasto com copos descartáveis total, e *ca* que representa o consumo de água.

Criação da rede social de profissionais do judiciário. Para melhorar o potencial do JusBD, a tabela RedeSocial foi criada, cujo objetivo é permitir o estudo dos relacionamentos entre profissionais do judiciário e respectivos salários. Semelhante a Batista *et al.* [2017] , tal rede social é modelada como um grafo ponderado $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, no qual $\mathcal{V}$ é o conjunto de nós que representam os profissionais do judiciário e $\mathcal{E}$ o conjunto de arestas não direcionadas que conectam profissionais que trabalham em um mesmo órgão

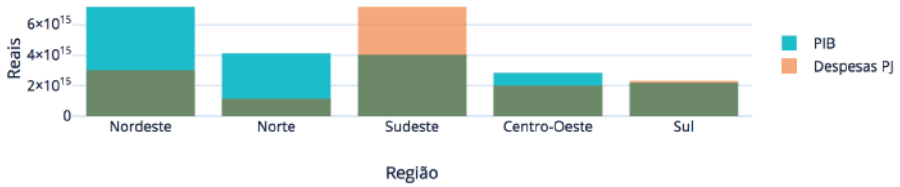


Figura 4. PIB (Produto Interno Bruto) versus despesas com o PJ (Poder Judiciário) por região brasileira no período de 2004 até 2017, ordenado decrescentemente pelo PIB. Tais dados foram extraídos da tabela EstatisticasPoderJudiciario. Note há sobreposição nas barras para melhor comparar o PIB com as Despesas do PJ.

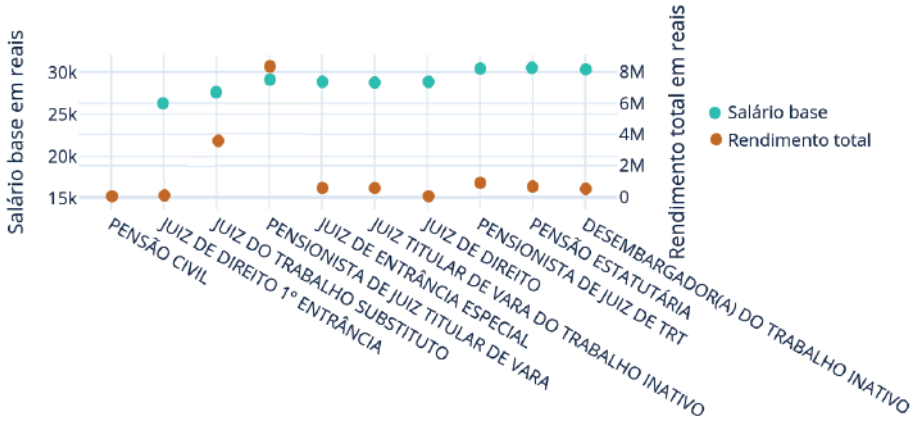


Figura 5. Top 10 salários base em reais por cargo do Poder Judiciário, com respectivo rendimento total. Esses dados foram obtidos por uma consulta na tabela ContraCheque.

do Poder Judiciário. Um exemplo é apresentado na Figura 3, nota-se que João, Ana, Flávia e Isis trabalham em um mesmo tribunal, ao passo que André e Maria trabalham em outro. Além disso, magistrados classificados como inativos não foram considerados na construção da rede social por não ter o período que ele trabalhou em tal tribunal.

4. Aplicações

O JusBD pode auxiliar qualquer estudo que precise consultar e/ou utilizar os dados do Poder Judiciário brasileiro, visto que um número grande de arquivos do CNJ são considerados. Por exemplo, investigar a desigualdade no judiciário brasileiro [Pereira and de Oliveira 2018] ao consultar os dados da tabela Socioambiental, ou analisar a racionalização do Poder Judiciário brasileiro [Tavares 2019] ao permitir o acesso às mudanças ocorridas no judiciário ao longo dos anos.

Em particular, a Figura 4 mostra uma análise prévia e simples que pode ser feita com o JusBD. Observa-se que as despesas com o Poder Judiciário é maior que o PIB nas regiões Sudeste e Sul. Isso chama a atenção para os gastos em tais regiões, que podem ser, por exemplo, com recursos humanos[12]. Adicionalmente, a Figura 5 mostra o Top 10 salários base de profissionais do Poder Judiciário, bem como o rendimento total dos

[12]O Globo - Gasto do Judiciário aumenta R$ 8 bilhões em 2017: `https://oglobo.globo.com/brasil/gasto-do-judiciario-aumenta-8-bilhoes-em-2017-aponta-relatorio-23014248`

mesmos. Em tal gráfico é possível analisar divergências muito grandes nos salários e verificar se ultrapassam o teto máximo, bem como obter evidências sobre rendimentos totais muito elevados.

5. Desafios e Limitações

No JusBD, há três limitações principais, descritas conforme segue.

Ausência de datas. Conforme apresentado na Figura 2, o JusBD possui doze tabelas e cinco delas possuem as colunas mês e ano. Entretanto, tais colunas referem-se ao período que tal dado foi inserido no site do CNJ e não é referente a cada linha na tabela. Dessa forma, não há datas sobre quando cada evento ocorreu no JusBD, por exemplo, não é possível saber quando um determinado contracheque foi gerado. Essa limitação também representa um desafio que é encontrar tais datas para atualizar o JusBD.

Rede social não temporal. Como não há datas no JusBD, não é possível construir uma estrutura de rede social temporal, por exemplo, conectando apenas profissionais do judiciário que trabalharam no mesmo período de tempo. Entretanto, estamos trabalhando para considerar tal período e melhorar a construção da rede.

Falta de dados sobre processos públicos. Outra limitação do JusBD é a falta de dados sobre processos públicos. Por enquanto, é possível apenas extrair informações da parte financeiro e socioambiental do Poder Judiciário. No entanto, a inclusão de dados de processos é um plano para trabalhos futuros.

Finalmente, um dos principais desafios para construção do JusBD é lidar com bases de dados em diferentes formatos e não padronizados. Isso dificulta a extração automatizada e integração de tais dados no JusBD.

6. Conclusões

Neste artigo, foi apresentado o JusBD, um banco de dados construído com dados do Poder Judiciário brasileiro, os quais foram extraídos do site do CNJ. O JusBD integra dados sobre a remuneração dos magistrados, estatísticas oficiais e balanço socioambiental do Poder Judiciário, o que possibilita diferentes análises, principalmente, forense. O JusBD também possibilita a realização de análises das interações sociais entre profissionais do judiciário para, por exemplo, verificar relacionamentos entre profissionais envolvidos em processos. Finalmente, o banco de dados proposto neste trabalho é fácil de usar e permite a rápida consulta a dados que estavam em arquivos separados.

Em trabalhos futuros, planejamos incluir dados de processos públicos, e incrementar a tabela rede social com dados coletados de mídias e redes sociais online. Assim, será possível lançar versões mais completas do JusBD. Também pretendemos utilizar os dados para fazer análises forense relevantes.

Agradecimentos. Trabalho parcialmente financiado pelo Programa Institucional de Bolsas de Pesquisa - IFMG e pela Capes.

Referências

Al Mutawa, N., Baggili, I., and Marrington, A. (2012). Forensic analysis of social networking applications on mobile devices. *Digital Investigation*, 9:S24–S33.

Anderson, S. and Williams, T. (2018). Cybersecurity and medical devices: Are the iso/iec 80001-2-2 technical controls up to the challenge? *Computer Standards & Interfaces*, 56:134–143.

Arshad, H., Jantan, A., and Omolara, E. (2019). Evidence collection and forensics on social networks: Research challenges and directions. *Digital Investigation*.

Artz, D. (2001). Digital steganography: hiding data within data. *IEEE Internet computing*, 5(3):75–80.

Batista, N. A., Alves, G. B., Gonzaga, A. L., and Brandão, M. A. (2017). Gitsed: Um conjunto de dados com informações sociais baseado no github. In *Procs. of SBBD-Dataset Showcase Workshop*, pages 224–233.

Becker, I., Hutchings, A., Abu-Salma, R., Anderson, R. J., Bohm, N., Murdoch, S. J., Sasse, M. A., and Stringhini, G. (2017). International comparison of bank fraud reimbursement: customer perceptions and contractual terms. *Journal of Cybersecurity*, 3(2):109–125.

Breitinger, F. and Roussev, V. (2014). Automated evaluation of approximate matching algorithms on real data. *Digital Investigation*, 11:S10–S17.

Dadkhah, M., Lagzian, M., and Borchardt, G. (2018). Identity theft in the academic world leads to junk science. *Science and engineering ethics*, 24(1):287–290.

Greengard, S. (2012). On the digital trail. *Communications of the ACM*, 55(11):19–21.

Guan, H., Kozak, M., Robertson, E., Lee, Y., Yates, A. N., Delgado, A., Zhou, D., Kheyrkhah, T., Smith, J., and Fiscus, J. (2019). Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *Procs. of IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72, Hawaii, USA.

Guarino, A. (2013). Digital forensics as a big data challenge. In *Procs. of ISSE securing electronic business processes*, pages 197–203. Springer.

Pereira, J. R. G. and de Oliveira, R. M. (2018). A (des) igualdade no judiciário brasileiro: breve comentário ao relatório"perfil sociodemográfico dos magistrados brasileiros", do conselho nacional de justiça. *Revista Publicum*, 4(2):214–219.

Roussev, V. (2011). An evaluation of forensic similarity hashes. *digital investigation*, 8:S34–S41.

Roussev, V. and Quates, C. (2013). File fragment encoding classification—an empirical approach. *Digital Investigation*, 10:S69–S77.

Tavares, L. C. A. (2019). A racionalização do sistema judicial no brasil: desafios e perspectivas. *Revista Brasileira de Sociologia do Direito*, 6(2).

Walnycky, D., Baggili, I., Marrington, A., Moore, J., and Breitinger, F. (2015). Network and device forensic analysis of android social-messaging applications. *Digital Investigation*, 14:S77–S84.

Zainudin, N. M., Merabti, M., and Llewellyn-Jones, D. (2010). A digital forensic investigation model for online social networking. In *Procs. of the 11th annual conference on the convergence of telecommunications, Networking & Broadcasting*, pages 21–22, Liverpool, Inglaterra.

MusicOSet: An Enhanced Open Dataset for Music Data Mining

Mariana O. Silva, Laís M. Rocha, Mirella M. Moro

Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

{mariana.santos,laismota,mirella}@dcc.ufmg.br

Abstract. *We present the* MusicOSet*, an open and enhanced dataset of musical elements (music, albums, and artists) suitable for music data mining. We describe the creation process and the data contents, along with usage examples and possible applications. The attractive features of* MusicOSet *include the enrichment of existing metadata and the popularity classification of the musical elements present in the dataset.*

1. Introduction

Individuals, organizations and governments are gradually realizing how the publication and availability of datasets can be useful. The fundamental role of datasets in several fields of research is irrefutable, especially for the initial progress of emerging topics and possibilities of experimental replications and thorough comparisons. For instance, public datasets are already an integral part of fields such as machine learning (Wine Quality [Cortez et al. 2009]), computer vision (ImageNet [Deng et al. 2009]), complex networks (SNAP [Leskovec and Krevl 2014]), social networks (GitSED [Batista et al. 2017]), digital libraries (DeduDLB [Silva and Brandão 2017]), biotechnology (MAMMOSET [Oliveira et al. 2017]) and Music Information Retrieval - MIR (MSD [Bertin-Mahieux et al. 2011]).

As in most scientific research fields, collecting and distributing datasets are important in MIR [Karydis et al. 2016]. Music Information Retrieval is a very important task in music data mining [Li et al. 2011]. Specifically, in such growing research domain, relevant musical content generally refers to audio files associated with lyrics, metadata and semantic information. While being a key piece in the progress of MIR research, the free distribution of such datasets and standardization are challenging tasks due to very restrictive copyright laws. However, to overcome these problems, many researchers follow an approach using free licenses (e.g., Creative Commons) [Goto et al. 2003; Defferrard et al. 2017] or just making acoustic feature vectors available [Bertin-Mahieux et al. 2011; Porter et al. 2015; Gemmeke et al. 2017], not audio data.

One issue in MIR is to apply multifaceted information from large musical databases for predicting hits. That and other issues evolved to a new research area called *Hit Song Science* (HSS), which aims to better understand the relationship between the intrinsic characteristics of the songs and their success. In other words, the goal is to predict whether a song offers the potential to become popular and commercially successful, thus reaching the top of the charts. In the MIR vision of HSS, the challenge is to gather a set of musical resources that can be mapped to music popularity. Once this mapping is ready, the process of predicting a new arbitrary song can be automated [Li et al. 2011].

There are different datasets in both MIR and HSS that cover a wide spectrum of the domain (see Table 1). However, none is directly applicable to extracting knowledge

of the popularity and intrinsic characteristics of musical elements (artists, songs and albums). Even worse, the main sources of music data extraction apply their respective track identification systems, making it challenging to collectively use multiple sources of musical data. Moreover, in most cases, there is no data available on less popular components. That is, the data collection contains only the popularity degree, with no information on the non-hits elements of the music industry.

To address the aforementioned challenges, we introduce the *MusicOSet*, an open and enhanced dataset of musical elements (artists, songs and albums) suitable for music data mining through the following features:

- Integration and centralization of different musical data sources;
- Calculation of popularity scores and classification of hits and non-hits musical elements, from 1962 to 2018;
- Enriched metadata for music, artists, and albums from the US popular music industry;
- Availability of acoustic and lyrical resources; and
- Unrestricted access in two formats (SQL database and compressed `.csv` files).

The remainder of this paper is organized as follows. In Section 2, we discuss related work. In Section 3, we describe the dataset, its creation processes as well as a detailed analysis of its content. We discuss how the data have been used and its applicability in Section 4. Next, we detail the potential challenges and limitations on using *MusicOSet* in Section 5 and conclude with future research directions in Section 6.

2. Related Work

There are numerous datasets publicly available that cover a broad spectrum in the music data mining area. These datasets provide plenty information related to music from different perspectives. However, most of them seek to provide content information (e.g., metadata, tags or acoustic features) by focusing on a particular purpose (e.g., recommendation systems, music information retrieval or music classification). Nevertheless, to embrace several tasks of music data mining, a dataset must provide a wide range of information in a centralized and easily accessible way, promoting the exploration of diverse musical aspects. Table 1 presents the most common datasets by comparing size (i.e., the total number of songs), metadata/acoustic features/lyrics/popularity data availability and release year, which is also the sorting field.

The RWC Music Database [Goto et al. 2003] is a copyrighted music database available specifically for search purposes. It was one of the first large-scale music databases containing six original components in different genres. However, the RWC size is currently considered small and it does not contain any further metadata. Another widely used dataset is the Computer Audition Lab 500-song (CAL500) [Turnbull et al. 2008]. CAL500 is a corpus of 500 tracks of songs chosen from a collection of western popular music authors. Each of the 500 songs was manually annotated by at least three people using a survey, with a total of $1,708$ musical annotations. Moreover, for each song, the dataset provides several features that have been extracted from audio files.

The Million Song Dataset (MSD) [Bertin-Mahieux et al. 2011] is perhaps one of the most used datasets in MIR. It provides audio features and metadata for one million contemporary popular music tracks. It stands out as one of the largest datasets currently available for research ends, totaling over 280 GB of data. Although MSD provides a great

Table 1: Comparison of the existing datasets

Dataset	Size	Metadata	Acoustic Features	Lyrics	Popularity Data	Year
RWC	365	yes	no	yes	no	2001
Cal500	500	no	yes	no	no	2007
MSD	1,000,000	yes	yes	no	no	2011
MusiClef	1,355	yes	yes	no	no	2012
TPD	23,385	yes	yes	no	yes	2014
Audio Set	2,084,320	yes	yes	no	no	2017
FMA	106,574	yes	yes	no	no	2017
HSPD	1,000,000	no	yes	no	yes	2019
MusicOSet	**20,405**	**yes**	**yes**	**yes**	**yes**	**2019**

deal of information, it is also criticized, mainly for the obscurity of the approaches used to extract content descriptors and the improbable integration of the different parts of the dataset. With a considerably reduced size, Schedl et al. introduced the MusiClef dataset [Schedl et al. 2013], a multimodal collection of professionally commented music. MusiClef includes editorial metadata, several audio features, annotation sets, collaboratively generated user tags, and MusicBrainz[1] identifiers to facilitate linking to other datasets.

Following a distinct approach to existing datasets, the Track Popularity Dataset (TPD) [Karydis et al. 2016] provides several sources of music popularity definition, within a period between 2004 and 2014. TPD also provides a mapping between different identification spaces, allowing the use of different data sources combined with metadata and contextual similarity information between tracks. More recently, the Hit Song Prediction Dataset (HSPD) based on the MSD was introduced [Zangerle et al. 2019]. With one million representative songs released between 1922 and 2011, the dataset also provides information about the MSD tracks that were included in the Billboard Hot 100 charts.

In a different perspective, Audio Set was introduced to bridge the gap in data availability between image and audio research [Gemmeke et al. 2017]. It is a large-scale dataset of hand-written audio events, which uses a carefully structured hierarchical ontology of 632 classes of literature-guided audio and manual curation. With a total of 2,084,320 songs, Audio Set exceeds MSD, becoming the largest set of music data. Concurrently, the Free Music Archive (FMA) was introduced as an open and easy-to-access dataset suitable to evaluate numerous music information retrieval (MIR) tasks [Defferrard et al. 2017]. The FMA consists of 343 days of audio and 917 GB, all under permissive Creative Commons licenses. It has complete metadata, including music title, album, artist and genres; user information, such as play counts, favorite items, and comments; along with high-quality audio files and some pre-calculated features.

As Table 1 shows, *MusicOSet* differs from all those datasets. It has more than 20 thousand songs, a regular size when compared to others. Nonetheless, what it lacks in number of songs, it makes up for in high quality information. In contrast to the datasets aforementioned, *MusicOSet* is the only one to provide all the attributes shown in Table 1.

[1] *MusicBrainz*: `https://musicbrainz.org/`

3. MusicSet

MusicOSet is an open and enhanced dataset of musical elements (artists, songs and albums) based on musical popularity classification. Provides a directly accessible collection of data suitable for numerous tasks in music data mining (e.g., data visualization, classification, clustering, similarity search, MIR, HSS and so forth). This section describes the entire creation and collection process, as well as its content, format and usage.

3.1. Creation Process

To create *MusicOSet*, the potential information sources were divided into three main categories: music popularity sources, metadata sources, and acoustic and lyrical features sources. Data from all three categories were initially collected between January and May 2019. Nevertheless, the update and enhancement of the data happened in June 2019.

Music Popularity Sources. Music popularity can be defined in different ways, including critics acclaim, social media and music platforms, sales profit, awards, etc. Another common approach is to rely on pop charts, such as the Billboard charts. To collect information of musical popularity, we used the *billboard.py*[2] Python API for access Billboard charts and perform the data collection. In total, we collected the last 56 years of the Hot 100 and Billboard 200 charts, ranging from 1962 (January 01, 1962) up to 2018 (December 31, 2018).

Metadata Sources. Subsequently, we used the Spotify, Genius, and Wikipedia platforms as sources of metadata and content. We choose these three web services since they all provide an API for research purposes. The *Spotipy*[3] library allows full access to all music data provided by the Spotify platform. For supplementary information, the *Wikipedia*[4] and *LyricsGenius*[5] libraries provide direct interfaces for accessing and analyzing data on Wikipedia and for songs, artists, and lyrics stored on Genius, respectively.

Acoustic and Lyrical Sources. To further enhance the *MusicOSet*, we included information on the lyrical and acoustic features of the collected songs. Acoustic fingerprints are condensed digital summaries of a song's phonic features [Ren et al. 2010]. These characteristics are the best measures available to capture the musical effect. That is, through acoustic features, the artistic style and creative experience are captured, characterizing the genome of a song. To collect information about the acoustic features of the songs from each album, we use the same *Spotipy* library. The fingerprints are produced by The Echo Nest[6], an online provider of musical intelligence that was acquired by Spotify in 2014. As for the lyrics information, we use the Python client for the Genius.com API, *LyricsGenius*.

Sources Integration. Determining when records are referring to the same real-world entity is essential in any Data Management effort that brings together data from multiple sources. This process is called *record linkage* and can be solved through probabilistic or fuzzy matching. Probabilistic text linkage is a very effective approach that uses string similarity functions, comparing two parts of the text and producing a single similarity metric. As

[2] *billboard.py*: `https://github.com/guoguo12/billboard-charts`
[3] *Spotipy*: `https://spotipy.readthedocs.io/`
[4] *Wikipedia*: `https://wikipedia.readthedocs.io/`
[5] *LyricsGenius*: `https://github.com/johnwmillr/LyricsGenius`
[6] *The Echo Nest*: `http://the.echonest.com/`

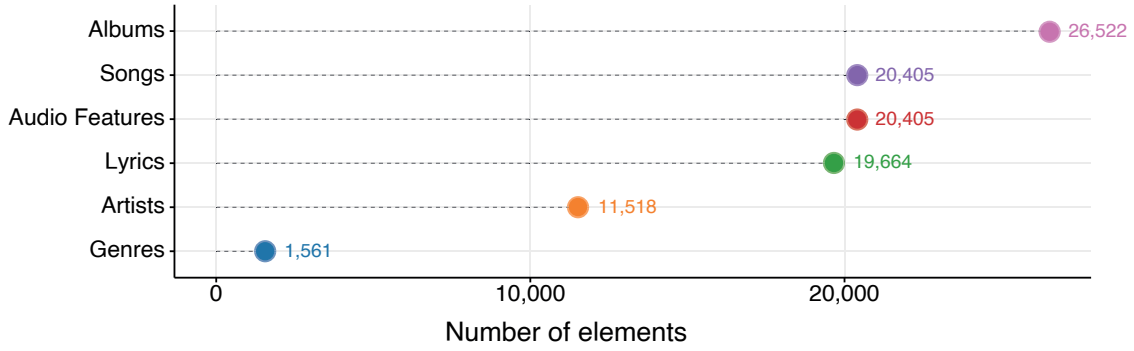


Figure 1: *MusicOSet* statistics

each data source has a different identification system, we used the *SequenceMatcher* class from the Python *difflib*[7] library, as well as the *Jaro-Winkler* algorithm from the *python-string-similarity*[8] library to map the music/artist/album records that refer to the same entity in all sources, with a similarity ratio of 0.9. However, because not all platforms incorporate information about all the gathered data, the mapping is not complete. In total, we were able to map approximately $82,5\%$ of the initial records collected.

3.2. Data Content

To facilitate the storage and visualization of data, we use a relational database management system (RDBMS) as a mechanism for storing the *MusicOSet* dataset. Figure 2 shows the database *schema*. Overall, it is composed of 13 tables that include the metadata/content, the acoustic and lyric features, and the popularity rating tables of the musical elements. Figure 2 also illustrates a division of the tables into three main segments: Popularity, Metadata and Song Features. Note that Popularity and Metadata are available in three different levels: artist (solo, duo or band), album (which is a collection of songs), and individual song. Such levels make the dataset more inclusive and easy to query, also enabling its use in different music data mining applications under varied aspects of the music industry. Figure 1 presents a quantitative description of the *MusicOSet* statistics.

Popularity. The popularity segment contains three tables (*ArtistChart*, *AlbumChart* and *SongChart*) consisting of ranking information collected from Billboard charts. Such tables include the inverted rating on a chart (*rank_score*); the highest rank achieved in any week of a year (*peak_position*); the number of weeks it has been on the charts in a year (*weeks_on_chart*); and the chart date. Specifically, the *SongChart* and *AlbumChart* tables were created from data collected from the Hot 100 and Billboard 200 charts, respectively. For creating the *ArtistChart* table, we weekly grouped the ranking information of the song/album artists featured in both previously mentioned Billboard charts. The other three remaining tables represent our popularity classification of the musical elements. Success may be measured by the presence of a song, album or artist on the ranking charts, such as Billboard. However, there is no equivalent for "unsuccessful songs" or "unpopular artists". With such restriction, there is no direct way to collect the unknown or less popular songs/albums/artists. To handle this limitation, we initially calculated a year-end score

[7] *difflib*: `docs.python.org/3.6/library/difflib.html`
[8] *python-string-similarity*: `github.com/luozhouyang/python-string-similarity`

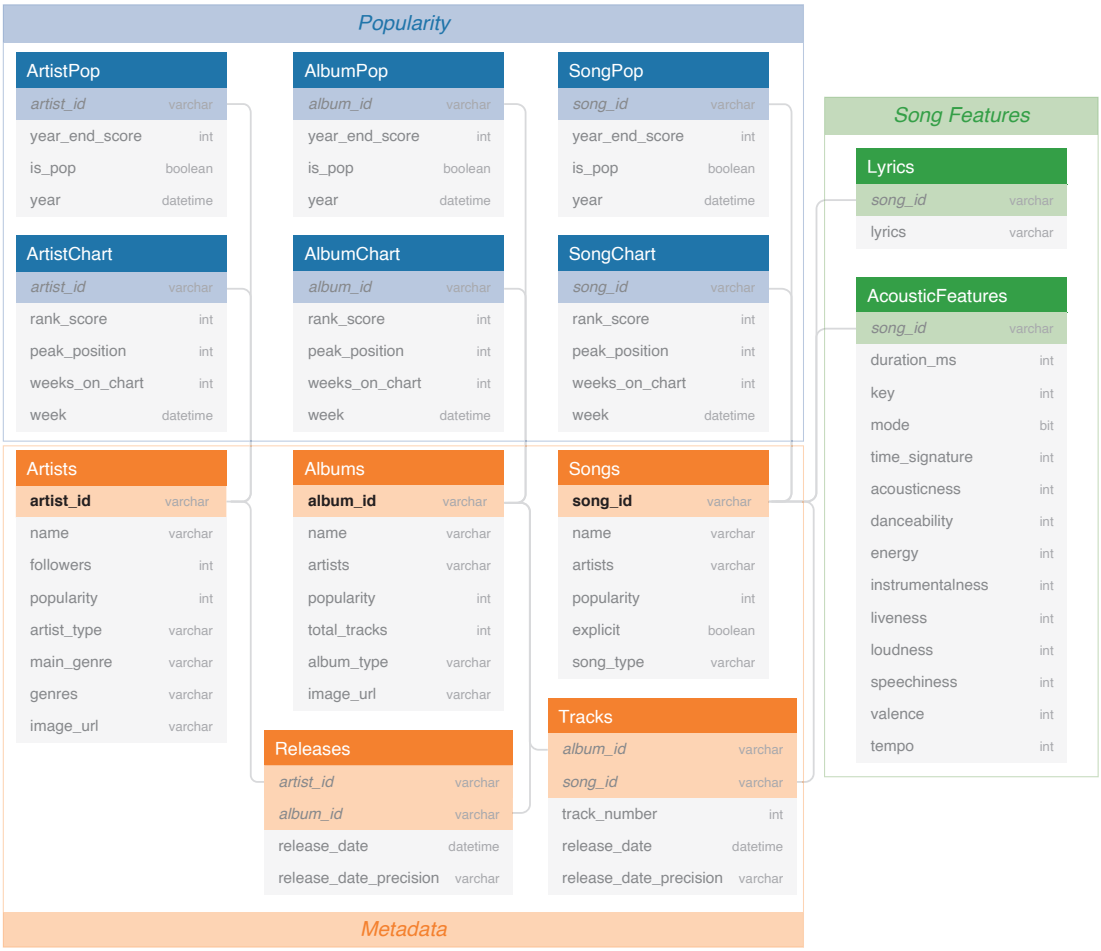


Figure 2: Schema for *MusicOSet*

that combines the scores of *peak_position* and *weeks_on_chart* and ranked the musical elements annually. Next, we assume that positive records (hits) correspond to the items that scored higher than the average for that year; whereas the negatives (non-hits) are the ones that obtained the lowest scores. Finally, we create a *boolean* field (*is_pop*), where *True* indicates a popular song/artist/album and *False*, the opposite.

Metadata. The metadata segment consists of textual and numeric information about songs, artists and albums. Basic information such as name, number of followers, popularity, and genre were collected directly from Spotify. The popularity field represents a value between 0 and 100, with 100 being the most popular. The song popularity is calculated by an algorithm and is based, in the most part, on the cumulative number of plays the track has and how recent those plays are. In other words, songs that are currently being played a lot receive higher popularity score than songs frequently played in the past only. Then, artist and album popularity are mathematically derived from song popularity. We also added information on types of song, artist and album. To capture the type of artists, we used the Wikipedia API to search for artist names and identify the presence of the terms “singer”, “band”, “duo”, “rapper” or “DJ”. As the type of songs, we distinguish only two types: solo songs (with only one artist present in its execution) or collaborative songs (where there is more than one artist). For the type of albums, we collected directly from Spotify,

which classifies the albums in three categories: album, single or compilation. To conclude, we also added URLs of artist images and album covers collected from both Spotify and Genius platforms. The remaining two tables (*Releases* and *Tracks*) were created to store album release information in the *Albums* table and song track information in the *Songs* table, respectively.

Song Features. Finally, the song features segment consists of only two tables: *Lyrics* and *AcousticFeatures*. The first table contains the lyrics of all songs present in the *Songs* table, which were collected using the *LyricsGenius* library. The second table contains acoustic fingerprints collected directly from Spotify[9]. Some acoustic fingerprints are objective (key, intensity, mode, tempo and time signature); others are more subjective (acousticness, danceability, energy, instrumentalness, liveness, speechiness and valence) and their values are calculated using The Echo Nest's music audio analysis tool [Jehan and DesRoches 2011]. We also consider the duration of the track as acoustic characteristics of a song.

3.3. Format and Usage

The *MusicOSet* is available including two separate parts: (A) a SQL file that creates the relational database and the 13 tables previously described (Section 3.1) and subsequently loads all data in the tables by a MySQL installation; (B) the same information as the tables in (A) but in `.csv` format to support fast use of the data and mitigate the need for a relational database. The full *MusicOSet* can be downloaded at `www.dcc.ufmg.br/~mirella/projs/bade`.

4. Applicability

A broad variety of music data mining tasks could be performed and analyzed using the *MusicOSet*. In this section, we share scenarios and possible applications, which help to illustrate the breadth and potential impact of the data available.

Metadata Analysis. One of the most direct applications for the dataset is the metadata analysis. Metadata analysis may involve, for example: music visualization, focusing on illustrating the metadata or acoustic contents; association mining, which refers to detecting correlations between different items in a set of data (e.g. acoustic characteristics, song lyrics, popularity, etc.); and clustering, which groups musical items into sets of similar objects based on their peer similarities. In addition to these tasks, other issues to explore over the dataset include: *How related are the artists? How have popular song lyrics changed over the years? Which musical genres have the highest average vocabulary? How different musical genres are correlated?* An example of a recent study that performs metadata analysis using *MusicOSet* is published in [Silva et al. 2019]. Through topological metrics and a clustering algorithm, the authors identified three well-defined communities with distinct collaboration patterns and notable discrepancies in levels of musical success. In addition, they found that successful artists are more likely to have profiles with a high degree of interaction and high diversification.

Hit Song Science. In the Hit Song Science (HSS) scenario, the main goal is to predict the success of songs before they are released. Therefore, researchers seek to identify features that make music more likely to be popular. There have been several notable studies on

[9]Spotify API Doc: `developer.spotify.com/documentation/web-api/reference/`

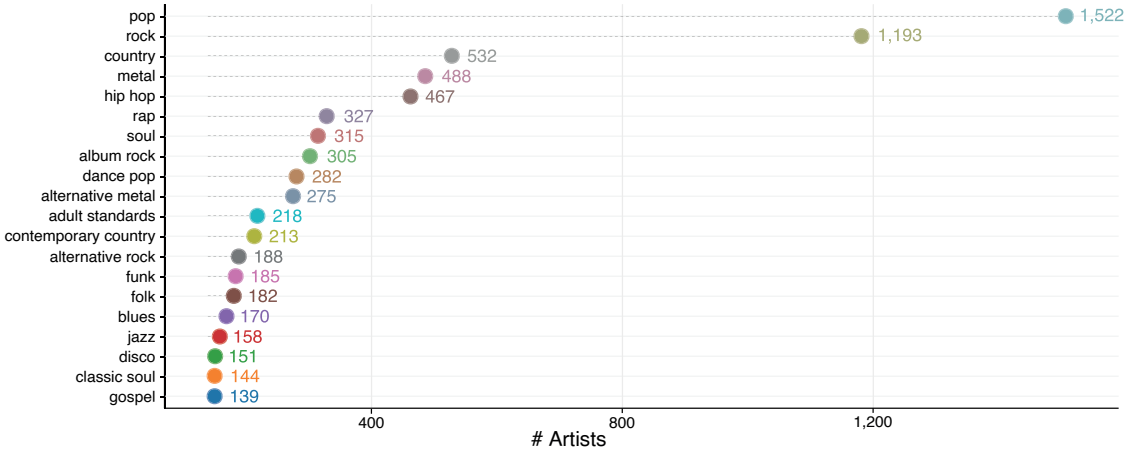


Figure 3: Distribution of artists in the 20 most common genres in *MusicOSet*

this topic, some of which focus on extracting acoustic and general lyric characteristics from songs, and then use standard classifiers to separate hits from non-hits. To study how the artists connect professionally can affect their musical success, Silva and Moro [2019] proposed a study using *MusicOSet* to assess whether there is a causal relationship between collaboration profiles and artist popularity.

Music Information Retrieval. Music Information Retrieval (MIR) is an emerging research area dedicated to meeting users' musical information needs [Li et al. 2011]. Musical recommendation and musical similarity are well explored issues in MIR due to the potential commercial value of a working system. The metadata and popularity information available on *MusicOSet* open up the possibility of a large-scale evaluation of song and music collaboration recommendation systems. Moreover, the available song features can assist the search for musical similarity. In principle, the user can provide an acoustic fingerprint set or a song lyric, and then the music search system will search for similar musical works based on the information provided by the user.

5. Limitations and Challenges

The *MusicOSet* is not free from limitations, which may be improved in future versions. The key challenges are related to the heterogeneity of the data sources used in the collection process. That is, due to the different identification systems, not all sources provide information about all the data gathered. Hence, some tables contain missing data. Another limitation is that the data sources consider only the mainstream and popular music, generalizing the information. Specifically, the data sources probably do not contain independent or less popular songs, artists or albums. Cultural and genre diversity is another issue: there is monopolization of US musical industry elements, as well as of pop and rock genres. This becomes explicit in Figure 3, which exposes the distribution of artists that have the 20 most common genres of the dataset.

In summary, although *MusicOSet* can be used to evaluate many tasks, some subsequent actions would further enhance the dataset. Additional features not considered in this first release, which are present in other datasets discussed in Section 2, could further enrich *MusicOSet*. For instance, the structure and content of the songs [Bertin-Mahieux et al. 2011], listener information [Schedl et al. 2013], extras artist metadata (e.g., related artists,

location, career time, etc.) or song metadata (e.g. track similarity, composer, publisher, genre, license, Spotify URL, etc.) [Karydis et al. 2016; Defferrard et al. 2017]. In addition, it would be critical to implement an automated Web-based collection and integration service that updates the dataset by capturing all sources.

6. Conclusion

This work introduces the *MusicOSet*: a cured, open and enhanced dataset of musical elements suitable for music data mining. Our contribution is related to integrating metadata, audio resources and musical popularity information. The *MusicOSet* is organized as a relational schema and made available in a public repository in two different formats. We also provide a statistical analysis of the dataset, as well as a discussion of the applicability and the main limitations in which the use of the *MusicOSet* involves. We believe that the dataset created along with the information described in this paper can be used for many music data mining tasks, such as MIR, classification, clustering, and prediction of successful songs (HSS).

Although *MusicOSet* remains two orders of magnitude behind the large-scale reference datasets analyzed here [Gemmeke et al. 2017; Bertin-Mahieux et al. 2011], for the best of our knowledge, it is the only one to provide a more complete set of attributes. In addition to providing enhanced metadata for songs, artists and albums from the US popular music industry, our dataset additionally makes available popularity scores, hit and non-hit ratings, as well as acoustic fingerprints and lyrics. All of this accessible in two formats (SQL database and compressed `.csv` files), with integration and centralization of different music data sources.

As future work, we plan to include new data sources to further expand and enrich *MusicOSet*, increasing the scope of potential applications. For instance, the added of different popularity sources, such as Last.fm and Spotify charts or even the number of Grammy Awards. Additionally, we plan to employ some traditional approaches to dealing with the missing data. A classic strategy would be to discard all elements for any sample that is missing one or more data components. The major problem with this method is the reduction of sample size. Therefore, we can alternatively fill in the missing data manually or input values using regression imputation. In the latter case, a regression model is estimated based on the observed values of variables to predict missing values. In other words, available information is utilized to predict the missing value of a specific variable.

Acknowledgments. The work is supported by CNPq, Brazil

References

Batista, N. A. et al. (2017). GitSED: Um Conjunto de Dados com Informações Sociais Baseado no GitHub. In *SBBD-Dataset Showcase Workshop*, pages 224–233.

Bertin-Mahieux, T. et al. (2011). The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval*, pages 591–596.

Cortez, P. et al. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553.

Defferrard, M. et al. (2017). FMA: A Dataset for Music Analysis. In *18th International Society for Music Information Retrieval Conference*.

Deng, J. et al. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Gemmeke, J. F. et al. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780.

Goto, M. et al. (2003). RWC music database: Music genre database and musical instrument sound database. *Proceedings of the 4th International Conference on Music Information Retrieval*, pages 229–230.

Jehan, T. and DesRoches, D. (2011). Analyzer documentation. *The Echo Nest*.

Karydis, I. et al. (2016). Musical track popularity mining dataset. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 562–572.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford Large Network Dataset Collection. `http://snap.stanford.edu/data`.

Li, T., Ogihara, M., and Tzanetakis, G. (2011). *Music Data Mining*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition.

Oliveira, P. H. et al. (2017). MAMMOSET: An enhanced dataset of mammograms. In *SBBD-Dataset Showcase Workshop*, pages 256–266.

Porter, A. et al. (2015). Acousticbrainz: a community platform for gathering music information obtained from audio. In *16th International Society for Music Information Retrieval Conference*.

Ren, L. et al. (2010). Dynamic Nonparametric Bayesian Models for Analysis of Music. *Journal of the American Statistical Association*, 105:458–472.

Schedl, M. et al. (2013). A Professionally Annotated and Enriched Multimodal Data Set on Popular Music. In *Proceedings of the 4th ACM Multimedia Systems Conference (MMSys 2013)*, Oslo, Norway.

Silva, M. O. and Brandão, M. A. (2017). Deduplicação de Nomes e Redes de Co-autoria na DBLP. In *SBBD-Dataset Showcase Workshop*, pages 203–212.

Silva, M. O. and Moro, M. M. (2019). Causality Analysis Between Collaboration Profiles and Musical Success. In *Proceedings of the 25th Brazilian Symposium on Multimedia and the Web*. [to appear].

Silva, M. O., Rocha, L. M., and Moro, M. M. (2019). Collaboration Profiles and Their Impact on Musical Success. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 2070–2077, Limassol, Cyprus. ACM.

Turnbull, D. et al. (2008). Semantic Annotation and Retrieval of Music and Sound Effects. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):467–476.

Zangerle, E. et al. (2019). Hit Song Prediction: Leveraging Low- and High-Level Audio Features. In *Proceedings of the 20th International Society for Music Information Retrieval Conference 2019 (ISMIR 2019)*. [to appear].

QualiSUS: um *dataset* sobre dados da Saúde Pública no Brasil

João Paulo Clarindo[1], **Wagner da Silva Fontes**[2], **Fábio Coutinho**[2]

[1]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo (USP)
São Carlos – SP – Brasil

[2]Instituto de Computação
Universidade Federal de Alagoas (UFAL)
Maceió – AL – Brasil

jpcsantos@usp.br, {wsf,fabio}@ic.ufal.br

Abstract. *In recent years, government agencies have increasingly made open data available. In Brazil, the Ministry of Health, by DATASUS, provides data from public health management systems. However, these datasets have proprietary formats and lack the use of a common schema pattern. Such issues make it difficult to manipulate data extracted from DATASUS systems, preventing their direct integration with external databases. In order to support researchers and managers, this paper presents QualiSUS, a dataset built with data from three DATASUS systems (SIHSUS, SIM and SINASC). QualiSUS aims to aggregate quality aspects by applying standardization techniques, suppressing invalid data and mapping to open formats such as JSON and CSV.*

Resumo. *Nos últimos anos, organizações governamentais têm crescentemente promovido a disponibilização de dados abertos. No Brasil, o Ministério da Saúde, através do DATASUS, disponibiliza dados relativos à gestão da saúde pública em escala municipal, estadual e nacional. Entretanto, essas bases de dados apresentam formatos proprietários e carecem do uso de padrões de representação. Tais fatores dificultam fortemente a manipulação dos dados extraídos dos sistemas DATASUS, impedindo sua integração direta com bases externas. Visando a apoiar pesquisadores e gestores, este trabalho apresenta QualiSUS, um* dataset *construído a partir de dados oriundos de três sistemas DATASUS (SIHSUS, SIM e SINASC). QualiSUS tem o objetivo de agregar aspectos de qualidade através da aplicação de técnicas de padronização, supressão de dados inválidos e mapeamento para formatos abertos tais como JSON e CSV.*

1. Introdução

Recentemente, diversas iniciativas que promovem a disponibilização de dados abertos governamentais têm surgido no mundo [Abella et al. 2019, Attard et al. 2015]. No Brasil, essa tendência tem sido verificada a partir de ações que buscam disponibilizar dados do governo para a sociedade tais como a criação da Lei de Acesso à Informação (Lei 12.527/2011) [BRASIL 2011], do Portal da Transparência[1] e do Portal de Dados Abertos[2].

[1]http://www.portaldatransparencia.gov.br
[2]http://dados.gov.br

Desde a década de 1990, o Ministério da Saúde possui um departamento responsável por manter bases de dados contendo informações da área da saúde – o DATASUS [DATASUS 2019c], que dispõe de mais de 200 sistemas que auxiliam na gestão e controle do SUS (Sistema Único de Saúde), destacando-se o SIM (Sistema de Informação sobre Mortalidade), SINASC (Sistema de Informações sobre Nascidos Vivos), SIA (Sistema de Informações Ambulatoriais) e SIHSUS (Sistema de Informação Hospitalar do SUS).

Os dados gerados por sistemas do DATASUS são oferecidos gratuitamente via servidor FTP (File Transfer Protocol)[3], sendo fornecido em formato proprietário ".DBC", uma versão comprimida do formato DBF (dBase database file), e utilizada primariamente pelo Sistema de Gerenciamento de Banco de Dados (SGBD) dBase [4]. Para a descompressão desses arquivos, o DATASUS disponibiliza as ferramentas TabWin e DBC2DBF [DATASUS 2019d], as quais foram desenvolvidas para executar em sistema operacional Windows e mostram-se limitadas para pesquisadores que almejam manipular um grande volume de dados. Por exemplo, em [de Lima et al. 2019], os autores ressaltam o esforço demandado para realizar uma análise qualitativa dos partos ocorridos em São Paulo e Mato Grosso, no âmbito do SUS, considerando os últimos 20 anos. Para tal, foi necessário percorrer os seguintes passos: (i) baixar diversos arquivos referentes ao SIHSUS; (ii) realizar a conversão desses arquivos para o formato DBF utilizando TabWin; (iii) persistir os dados convertidos em um SGBD (MySQL) e, finalmente, (iv) executar o processamento analítico sobre a base de dados persistida. Os passos verificados nesse trabalho representam uma necessidade comum a diversas outras iniciativas que se utilizam de dados providos pelos sistemas do DATASUS.

Além disso, cabe ressaltar que os dados disponibilizados pelo DATASUS, em alguns casos, utilizam padrões de representação distintos, dificultando a manipulação e integração de suas bases. Em uma mesma base de dados, inclusive, podem coexistir campos com padrões de representação diferentes. Outra característica que contribui para a diminuição da qualidade das bases do DATASUS é a presença de inconsistências e dados sem uso oriundos de sistemas legados [DATASUS 2016].

Todos esses fatores evidenciam que a manipulação de dados do DATASUS pode se tornar uma tarefa árdua. Visando apoiar a missão de pesquisadores e gestores que atuam na saúde pública, este trabalho apresenta QualiSUS, um *dataset* construído a partir de dados gerados por três sistemas do DATASUS (SIHSUS, SIM e SINASC), agregando aspectos de qualidade através da aplicação de técnicas de padronização, supressão de dados inválidos, eliminação de inconsistências e mapeamento para formatos abertos (JSON e CSV). Desta forma, o *dataset* produzido pode ser facilmente persistido em SGBDs (Relacionais ou NoSQL), ambientes de *Data Warehouse* e de mineração de dados.

Este documento está organizado da seguinte forma: a Seção 2 discute os trabalhos relacionados; a Seção 3 relata problemas encontrados nos dados oriundos do DATASUS; a Seção 4 descreve uma visão geral do QualiSUS, incluindo os formatos, abrangência e dicionários presentes; a Seção 5 explica em detalhes como QualiSUS foi construído; a Seção 6 apresenta algumas estatísticas sobre o *dataset* enquanto a Seção 7 discute sua aplicabilidade; finalmente, a Seção 8 traz as considerações finais acerca do trabalho, in-

[3] `ftp://ftp.datasus.gov.br/dissemin/publicos/`

[4] `www.dbase.com`

cluindo, suas limitações e trabalhos futuros, e a disponibilização do *dataset*.

2. Trabalhos Relacionados

Inúmeros trabalhos científicos fazem uso, em seus experimentos, de dados disponibilizados pelo DATASUS [de Lima et al. 2019, Souza and Freire 2014, Pereira et al. 2016, Clarindo and Coutinho 2014] e, normalmente, relatam o esforço envidado com a manipulação desses dados. Diante disso, foram desenvolvidos alguns trabalhos com propósito similar ao QualiSUS, os quais são discutidos a seguir.

[Petruzalek 2016] disponibiliza uma ferramenta, desenvolvida em R, para a leitura de arquivos diretamente em formato DBC, ou seja, sem a necessidade de utilização do TabWin para a conversão em formato DBF. O objetivo da ferramenta é simplificar o processo de importação de dados do DATASUS na plataforma de análise estatística R. Apesar da contribuição em superar o esforço da etapa de conversão para DBF, a solução proposta está restrita ao ambiente R, exigindo do pesquisador interessado algum nível de familiaridade e conhecimento prévio da linguagem R.

[Mendes et al. 2019] desenvolveram uma ferramenta que auxilia o pré-processamento e a visualização de dados do DATASUS. Trata-se de uma ferramenta web com módulos denominados *datasusImport* e *datasusViewer*. O primeiro é responsável por baixar arquivos DBC, transformá-los em DBF para carregamento em SGBD relacional enquanto que o último oferece a visualização dos dados carregados em gráficos. A ferramenta busca auxiliar o pesquisador na manipulação e visualização de dados do DATASUS, todavia, não foi disponibilizado acesso para a realização de testes com relação ao seu desempenho diante de alto volume de dados.

Similarmente aos trabalhos anteriormente mencionados, QualiSUS pretende auxiliar o pesquisador na manipulação de dados oriundos do DATASUS. Entretanto, apesar do intuito em comum, este trabalho segue uma abordagem distinta, com foco na construção de um *dataset* qualificado, em formato aberto, o que viabiliza seu uso e aplicação de modo irrestrito conforme interesse do usuário. Desta forma, QualiSUS possibilita ao pesquisador maior abrangência no escopo do uso de ferramentas para análise, visualização e persistência de seus dados. Assim, a partir da disponibilização nos formatos CSV e JSON, a manipulação dos dados poderá facilmente considerar a utilização de SGBDs (Relacionais ou NoSQL), ferramentas ETL (*Extraction, Transform and Load*), ambientes de nuvem, ferramentas de análise tais como R e Weka. Além disso, cabe destacar o aspecto qualitativo da proposta, que busca unificar o uso de padrões comuns para as diferentes bases históricas do DATASUS as quais encontram-se com diferentes padrões utilizados no decorrer do tempo (1979 – 2017), são os casos dos padrões CID-9/CID-10, Código identificador de municípios 6-dígitos/7-dígitos e normalização dos padrões de dados.

3. Problemas encontrados nos dados do DATASUS

Os dados gerados pelos sistemas do DATASUS encontram-se em formato DBC, disponíveis via FTP e organizados por Unidade Federativa e mês e/ou ano de geração. Uma análise minuciosa sobre esses dados permitiu a verificação de problemas que comprometem a qualidade dos dados. Nesta seção são discutidos alguns desses problemas:

- Em alguns casos, os **campos nulos** são representados por uma sequência de caracteres ”0”. No informe técnico dos dados do SIHSUS, por exemplo, existem 20 campos descritos como “zerados” [DATASUS 2016]. Esta prática aumenta o tamanho da base de dados, dificultando a sua manipulação à medida que exige mais recursos do pesquisador interessado em analisar seus dados.
- O DATASUS **utiliza dois padrões para identificar um município**, um padrão de código composto por seis dígitos, sem um dígito de controle e outro com sete dígitos, incluindo dígito de controle [DATASUS 2019a]. Nas bases do DATASUS, não existe um padrão único para a identificação dos municípios brasileiros, podendo ocorrer, inclusive, o uso de ambos os padrões simultaneamente na mesma base. Sendo assim, caso um pesquisador necessite integrar dados referentes ao SIM e ao SIHSUS, por exemplo, deverá utilizar mecanismos de conversão destes códigos para viabilizar a visão integrada desses dados
- Nas bases do DATASUS são encontrados padrões de CID distintos. O CID (Classificação Internacional de Doenças) é uma classificação e codificação de doenças desenvolvida pela OMS (Organização Mundial de Saúde) [OMS 2019]. O sistema SIM utiliza código CID para registrar a causa da morte. Entretanto, **dados do SIM anteriores ao ano de 1996 adotam o padrão CID-9, enquanto que a partir de 1996 em diante adotam o padrão CID-10**. Desta forma, para proceder, por exemplo, uma análise temporal mais abrangente considerando uma determinada causa de morte, o pesquisador precisará realizar conversões entre os códigos CID.
- Nas bases do DATASUS, **as datas não seguem um padrão de representação comum**: há casos que utilizam o padrão ISO 8601, padrão internacional para representação de datas [ISO 2019], ou que utilizam o padrão NBR 5892 da ABNT (Associação Brasileira de Normas Técnicas) [ABNT 1989]. Também foram encontrados casos em que os campos não seguem nenhum destes padrões, dificultando a análise temporal das bases.
- Em todas as bases de dados verificadas neste trabalho foram identificadas a presença de **campos não-preenchidos**, o que diminui o potencial dos resultados gerados pelas análises dos dados.
- As estruturas dos arquivos oferecidos por um sistema do DATASUS não oferecem um **padrão comum para os campos**. Essas estruturas são baseadas nos formulários que alimentam o sistema. Por exemplo, o SINASC, que se baseia na Declaração de Nascidos Vivos, disponibiliza três padrões diferentes (antes de 1996, 1996-2015 e 2015-atual). Caso o pesquisador deseje consultar um determinado campo, deve-se considerar que podem existir outros campos com a mesma informação, mas com rótulo diferente.

4. Visão Geral do QualiSUS

No intuito de agregar mais qualidade aos dados da saúde pública, superando alguns dos problemas apresentados na Seção 3, foi desenvolvido o *dataset* QualiSUS, com uso de padrão de representação comum e disponibilização em formatos abertos. O *dataset* reúne dados do SIHSUS, que contém registros dos atendimentos provenientes de internações hospitalares, do SIM, que contém registros sobre a mortalidade no país, e do SINASC, que contém registros sobre nascimentos [DATASUS 2019b]. Nas subseções seguintes são apresentadas as principais características do *dataset* QualiSUS.

4.1. Formato e Abrangência

O QualiSUS está disponível em dois formatos abertos e regulamentados pelo IETF (*Internet Engineering Task Force*): CSV (*Comma-Separated Values*), que são arquivos de texto com formato tabular, e que utiliza vírgulas como separadores [Shafranovich 2005]; e JSON (*JavaScript Object Notation*), um formato de padrão aberto cuja estrutura é simples de ser lida tanto por humanos como por máquinas [Bray 2017]. Ambos os formatos são amplamente utilizados: o CSV é suportado por inúmeros SGBDs , enquanto o JSON é utilizado em APIs (Application Programming Interface) e SGBDs NoSQL [Drapeau 2018]. Os dados disponibilizados pelo QualiSUS abrangem os seguintes períodos de tempo:

- SIHSUS: 2008 até 2017
- SIM: 1979 até 2017
- SINASC: 1996 até 2017

4.2. Definição de campos e dicionário

Para definir o esquema utilizado em QualiSUS, foram levados em consideração os formulários que alimentam as bases de dados do DATASUS: Declaração de Nascidos Vivos (SINASC), Declaração de Óbito (SIM) e o Laudo para Solicitação de Autorização de Internação Hospitalar (SIHSUS). Em cada documento, os campos são organizados em blocos. Por exemplo, a Declaração de Óbito possui blocos sobre o local de ocorrência do óbito, identificação do falecido, condições e causas do óbito e um bloco específico para óbitos fetais e de menores de um ano, com informações sobre a mãe da criança e sua gestação. [Brasil Ministério da Saúde 2009]

QualiSUS procura manter sua estrutura JSON similar à estabelecida nesses documentos. Uma das vantagens em buscar essa similaridade é facilitar a compreensão da semântica dos campos. A Figura 1 mostra o esquema do QualiSUS para os dados oriundos do SIM, o qual mantém a estrutura original dos blocos da Declaração de Óbito.

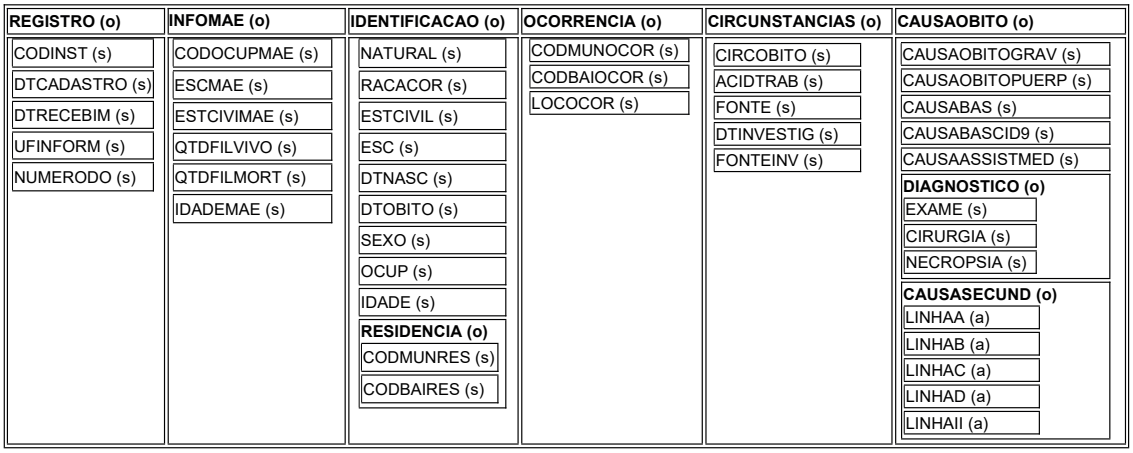
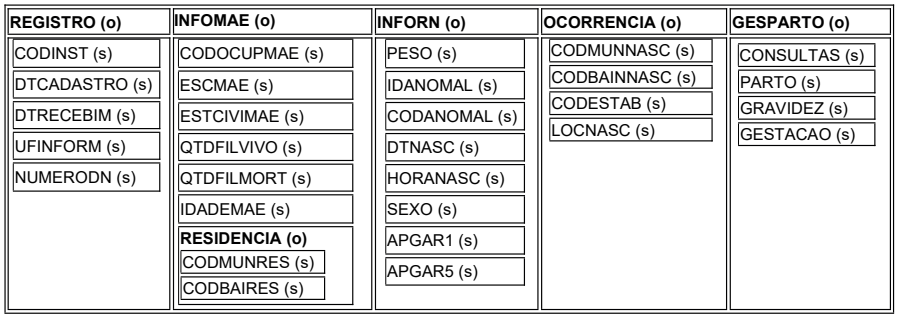


Figura 1. Estrutura do QualiSUS para o SIM baseado na Declaração de Óbito

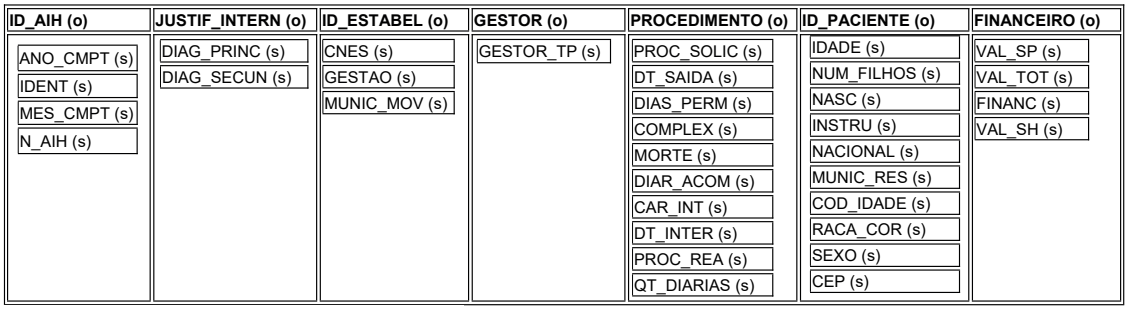
Campos multi-valorados presentes nos documentos, por exemplo, o campo "Linha II" da Declaração de Óbito (lista de causas secundárias relativas ao óbito), estão disponíveis na forma de JSONArray (conforme visto na Figura 1), facilitando as consultas para determinados tipos de causas para óbito. Disposições similares também foram

feitas nas bases do SINASC e SIHSUS representadas pelas Figuras 2 e 3, respectivamente. O dicionário com todos os detalhes a respeito da disposição dos campos no QualiSUS, incluindo suas descrições, encontra-se disponível junto ao *dataset* (detalhes da disponibilização do dicionário na Seção 8). Arquivos auxiliares para conversão de códigos do CID-10 ou do CBO (Classificação Brasileira de Ocupações) também se encontram disponíveis.



Legenda: (o) Objeto; (s) String; (a) Array

Figura 2. Estrutura do QualiSUS para o SINASC baseado na Declaração de Nascidos Vivos



Legenda: (o) Objeto; (s) String; (a) Array

Figura 3. Estrutura do QualiSUS para o SIHSUS baseado no Laudo para Solicitação de Autorização de Internação Hospitalar

5. Construção do Dataset QualiSUS

Para construir o *dataset* QualiSUS, foi desenvolvida uma aplicação na linguagem Python com três módulos: módulo de extração, módulo de transformação e módulo de consolidação. A Figura 4 apresenta um esquema geral da aplicação. As subseções a seguir detalham o funcionamento de cada módulo.

5.1. Módulo de Extração

O módulo de extração é responsável por extrair todos os arquivos DBC relativos ao SIHSUS, SIM e SINASC a partir do servidor FTP do DATASUS e convertê-los para o formato DBF, que são lidos utilizando a biblioteca *dbf*[5]. A conversão DBC para DBF é feita pela ferramenta dbc2dbf, disponibilizada pelo DATASUS. Por fim, os arquivos DBF gerados são enviados para o módulo de transformação.

[5]https://pypi.org/project/dbf/

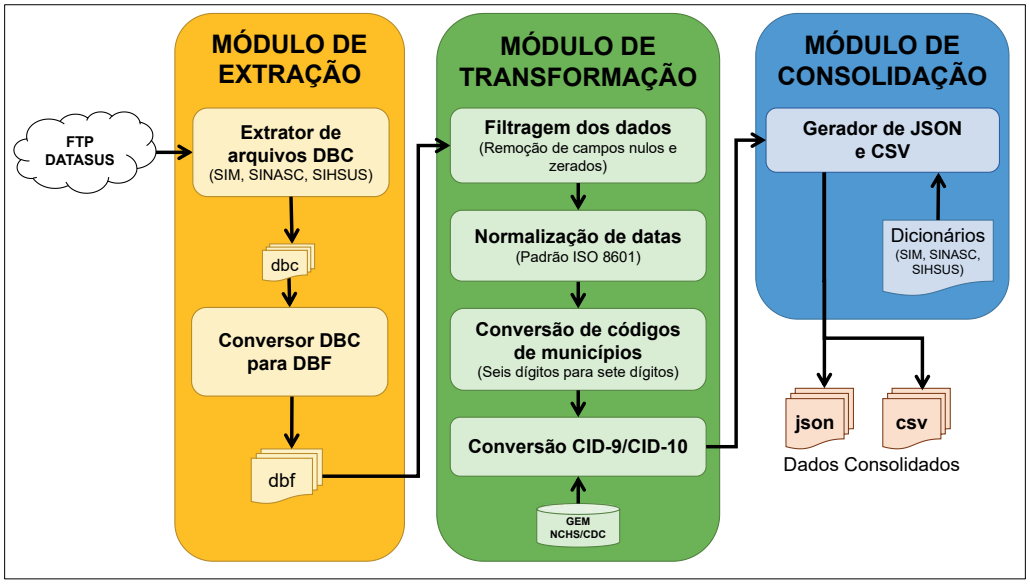


Figura 4. Esquema da aplicação para coleta dos dados e formação do QualiSUS

5.2. Módulo de Transformação

O módulo de transformação é responsável por selecionar os campos válidos, ignorando os campos nulos e zerados, e realizar outras operações que são descritas a seguir:

- A **conversão de códigos de municípios** foi implementada no módulo de transformação uma vez que os dados do DATASUS utilizam dois formatos distintos para códigos relacionados a municípios, 6-dígitos e 7-dígitos. Utilizando uma tabela de conversão disponibilizada por [DATASUS 2019a], os campos de municípios foram todos transformados para o padrão 7-dígitos.
- Considerando a **conversão de CID-9/CID-10**, conforme mencionado na Seção 3, o campo da causa do óbito, relativo aos dados do SIM anteriores a 1996, utilizam o padrão CID-9, enquanto os dados a partir de 1996 fazem uso do padrão CID-10. Essa representação distinta exige uma conversão para um padrão comum quando for necessária a obtenção de dados que compreendam um período mais abrangente (ex. década de 1990). Em QualiSUS, CID-10 foi o padrão de representação comum escolhido, sendo implementada a conversão do padrão CID-9 para CID-10 através do GEM (*General Equivalence Mapping*), disponibilizado pelo Centro Nacional para Estatísticas de Saúde (*National Center for Health Statistics* – NCHS/CDC), órgão associado ao Departamento de Saúde dos Estados Unidos [Ross-Davis 2012]. Além disso, foi adicionado um campo extra denominado CAUSABASCID9 para guardar o valor original nas tuplas que utilizavam o padrão CID-9 antes da conversão para CID-10. Faz-se necessária tal conduta, pois, em alguns casos, pode não haver no GEM a devida correspondência do valor em CID-9 para CID-10 [Schulz et al. 1998], logo, não haverá conversão e o pesquisador disporá apenas da informação legada representada no campo CAUSABASCID9.
- O módulo de transformação realiza a **conversão de formatos de datas**, determinando em qual formato a data encontra-se disponível, e realizando a conversão para o padrão ISO 8601, padrão internacional de descrição de datas [ISO 2019].

5.3. Módulo de Consolidação

Após a transformação dos dados, os campos são consolidados em arquivos JSON e CSV. A disposição dos campos está organizada conforme os dicionários exibidos nas Figuras 1, 2 e 3. Dados com disposição de campos distintos (como o SIM), são normalizados para os campos equivalentes do dicionário e os arquivos são organizados por unidade federativa.

6. Estatísticas dos dados

Nas subseções seguintes são apresentados alguns dados estatísticos baseados no *dataset* QualiSUS.

6.1. SIHSUS

A Figura 5 exibe um gráfico do número de registros no SIHSUS por regiões considerando o período entre janeiro/2008 e dezembro/2017. O gráfico mostra que o maior volume de dados sobre internações do SUS são relativos às regiões Sudeste e Nordeste do país.

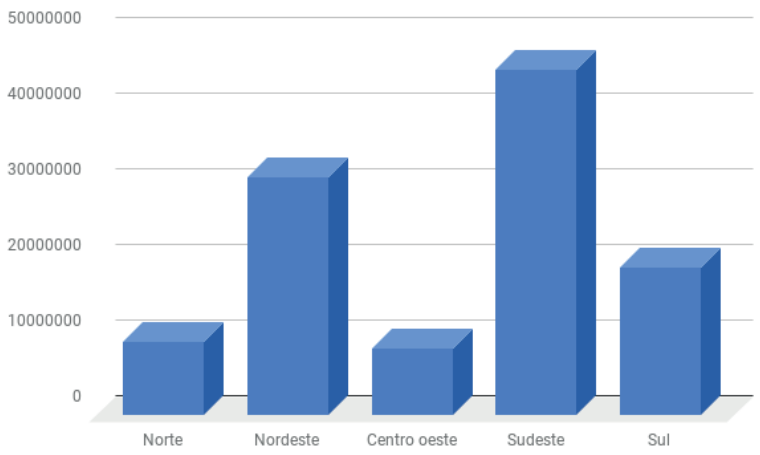


Figura 5. Quantidade de registros de internações no SIH por região entre janeiro de 2008 e maio de 2017 (em milhões)

6.2. SIM/SINASC

O gráfico mostrado na Figura 6 refere-se à quantidade de nascidos vivos e óbitos na região Nordeste entre os anos de 1996 e 2017. Conforme visualizado no gráfico, a quantidade de óbitos mostrou-se crescente, enquanto a quantidade de nascidos vivos tem decrescido levemente.

7. Aplicações do QualiSUS

O *dataset* QualiSUS pode ser utilizado por profissionais da saúde, pesquisadores, gestores e analistas de dados, aplicados em trabalhos com abordagens distintas sobre temas relacionados à saúde pública, epidemiologia, indicativos socieconômicos, etc. Os trabalhos que mais podem tirar proveito do QualiSUS são aqueles que necessitam realizar análises de dados em ambientes cujas bases originárias do DATASUS não são suportadas, devido sua disponibilização em formatos proprietários e pouco acessíveis. A representação em formatos abertos e o uso de padrões comuns, permite ao QualiSUS seu carregamento direto

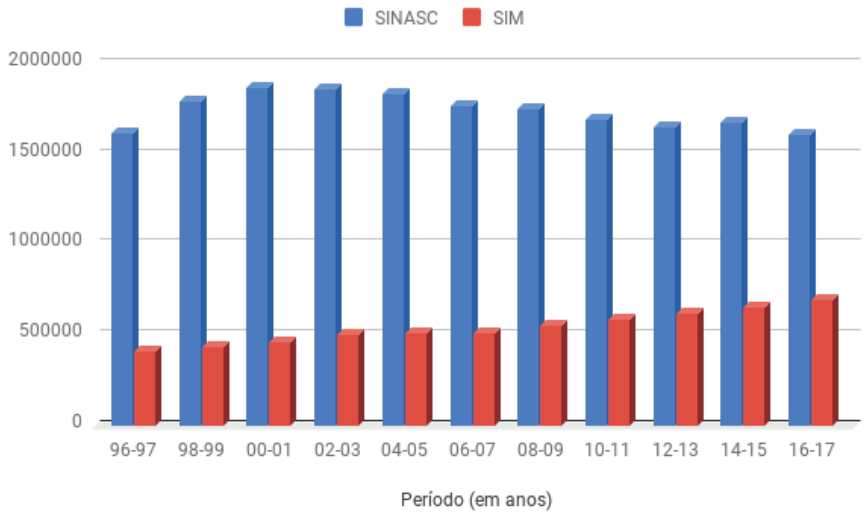


Figura 6. Quantidade de registros de nascidos vivos e óbitos, em SIM e SINASC, na região Nordeste (por biênio)

para diversos ambientes de análise e manipulação de dados tais como R, Weka, SGBDs Relacionais, NoSQL, ambientes de *data warehouse* e aplicações semânticas.

Além disso, por QualiSUS ser disponibilizado em formato aberto, permite ao pesquisador, facilmente, integrá-lo com bases de dados externas. Por exemplo, com o intuito de verificar qual o grau de relação existente entre a quantidade de internações relacionadas a doenças causadas por protozoários (como malária e amebíase) e a condição do sistema de saneamento básico em uma cidade, pode-se integrar os dados disponibilizados pelo QualiSUS com dados de saneamento básico de cidades, que estão disponíveis no Portal Brasileiro de Dados Abertos[6] nos formatos JSON e CSV.

8. Considerações Finais

Este artigo apresentou QualiSUS, um *dataset* construído a partir de dados oriundos da plataforma DATASUS com o objetivo de agregar qualidade através da padronização e supressão de dados inválidos e mapeamentos para os formatos abertos JSON e CSV. Esse diferencial qualitativo quando comparado aos dados presentes nos sistemas DATASUS permite ao *dataset* QualiSUS prover um ambiente onde pesquisadores, gestores e demais interessados possam facilmente utilizar ferramentas para a análise dos dados de mortalidade, nascidos vivos e internações.

Apesar de apresentar um período amplo de abrangência, pode-se considerar como limitação do *dataset* o fato dos dados finalizarem no ano de 2017, todavia, cabe salientar que os sistemas SIM e SINASC são disponibilizados somente até o ano de 2017, não havendo dados referentes aos anos de 2018 e 2019. Similarmente, o SIHSUS é disponibilizado a partir do ano de 2008, pois não há documentação disponível no servidor FTP do DATASUS para dados anteriores a 2008. Eventualmente, pode haver inconsistências inerentes à base de dados original e que sejam de verificação difícil, demandando, assim, a utilização de ferramentas mais avançadas para eliminá-las.

[6] http://dados.gov.br/dataset?tags=saneamento

Em trabalhos futuros, pretende-se expandir o *dataset* a partir da inclusão de dados do SIHSUS referente ao período compreendido entre os anos 1992 e 2007 (mediante a obtenção da documentação necessária). Cabe ressaltar que a aplicação desenvolvida para a geração dos *datasets* do QualiSUS é facilmente adaptável para a inclusão de outras bases oferecidas pelo DATASUS, tais como SIASUS (Sistemas de Informações Ambulatoriais do SUS), que compreende dados acerca de atendimentos ambulatoriais, e SISPRENATAL (Sistema de Acompanhamento da Gestante), que fornece informações sobre gestantes.

O QualiSUS encontra-se disponível em `https://github.com/jpclarindo/QualiSUS`. O *dataset* compreende dados do SIHSUS, SIM e SINASC, divididos por unidade federativa, ficando a critério do pesquisador a melhor opção para seu estudo de caso. Documentação relativa aos campos e tabelas adicionais também são disponibilizadas.

Referências

Abella, A., Ortiz-de Urbina-Criado, M., and De-Pablos-Heredero, C. (2019). The process of open data publication and reuse. *Journal of the Association for Information Science and Technology*, 70(3):296–300.

ABNT (1989). NBR 5892:1989 Indication of dates. Technical report, ABNT - Associação Brasileira de Normas Técnicas, Brasília.

Attard, J., Orlandi, F., Scerri, S., and Auer, S. (2015). A systematic review of open government data initiatives. *Government Information Quarterly*, 32(4):399–418.

BRASIL (18 de novembro de 2011). *Lei nº 12.527/2011*. Brasil. Casa Civil, Brasília. Diário Oficial da União, 16 de maio. 2012.

Brasil Ministério da Saúde, Conselho Federal de Medicina, C. B. d. C. d. D. (2009). *A declaração de óbito: documento necessário e importante*, volume 1. Editora MS.

Bray, T. (2017). The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259, RFC Editor.

Clarindo, J. P. and Coutinho, F. (2014). IGOV: um Sistema de Integração de Dados Governamentais. *Revista Brasileira de Administração Científica*, 5(2):8–16.

DATASUS (2016). Disseminação de Informações do Sistema de Informações Hospitalares (SIH) - Informe Técnico referente ao processamento 2016-03. `ftp://ftp.datasus.gov.br/dissemin/publicos/SIHSUS/200801_/Doc/IT_SIHSUS_1603.pdf`. [Online; acesso em jul. 19].

DATASUS (2019a). Cadastros Nacionais. `http://datasus.saude.gov.br/noticias/atualizacoes/59-sistemas-e-aplicativos/cadastros-nacionais?start=15`. [Online; acesso em jul. 19].

DATASUS, M. d. S. (2019b). DATASUS – Sistemas e Aplicações. `http://datasus.saude.gov.br/sistemas-e-aplicativos`. [Online; acesso em jul. 19].

DATASUS, M. d. S. (2019c). Sobre o DATASUS. `http://datasus.saude.gov.br/datasus`. [Online; acesso em jul. 19].

DATASUS, M. d. S. (2019d). TABWIN. http://datasus.saude.gov.br/projetos/10-informacoes-de-saude/155-tabwin. [Online; acesso em jul. 19].

de Lima, C. A., Mendonça, S., and de Lima, P. (2019). Partos no Sistema Único de Saúde em dois Estados do Brasil: uma análise qualitativa em três momentos dos últimos 20 anos. *CIAIQ2019*, 2:1029–1037.

Drapeau, M. (2018). The State of CSV and JSON. https://medium.com/@martindrapeau/the-state-of-csv-and-json-d97d1486333. [Online; acesso em jul. 19].

ISO (2019). ISO 8601-1:2019: Date and time – Representations for information interchange. Standard, International Organization for Standardization, Geneva, CH.

Mendes, D., Lobato, F., and Jacob Jr, A. (2019). Ferramenta de Pré-Processamento e Visualização de dados do DATASUS. In *Anais do VII Workshop de Transparência em Sistemas – WTranS*, pages 1–10. SBC.

OMS (2019). International Classifications of Diseases (ICD). https://www.who.int/classifications/icd/en/. [Online; acesso em jul. 19].

Pereira, F. J. R., da Silva, C. C., and Neto, E. d. A. L. (2016). Sensitive primary conditions brazilian panorama in 2013. *Journal of Nursing UFPE/Revista de Enfermagem UFPE*, 10(7).

Petruzalek, D. (2016). READ. DBC: um pacote para importação de dados do DATASUS na linguagem R. *J. health inform*, 8(supl. I):601–605.

Ross-Davis, S. V. (2012). Preparing for ICD-10-CM/PCS: one payer's experience with general equivalence mappings (GEMs). *Perspectives in Health Information Management/AHIMA, American Health Information Management Association*, 9(Winter).

Schulz, S., Zaiss, A., Brunner, R., Spinner, D., and Klar, R. (1998). Conversion Problems concerning Automated Mapping from ICD-10 to ICD-9. *Methods of Information in Medicine*, 37(03):254–259.

Shafranovich, Y. (2005). Common Format and MIME Type for Comma-Separated Values (CSV) Files. RFC 4180, RFC Editor.

Souza, R. and Freire, S. (2014). Integração de dados ambulatoriais de quimioterapia e radioterapia registrados nas bases de dados do sus. In *Anais do XXIV Congresso Brasileiro de Engenharia Biomédica – CBEB*.

SMMnet: A Social Network of Games Dataset

Leonardo Mauro Pereira Moraes[1], **Robson Leonardo Ferreira Cordeiro**[1]

[1]Institute of Mathematics and Computer Sciences, University of Sao Paulo
Av. Trabalhador Sancarlense, 400, Sao Carlos, SP, Brazil

leonardo.mauro@usp.br, robson@icmc.usp.br

Abstract. *Online games have become a popular form of entertainment, reaching millions of players. These players produce many types of interactions with the games, e.g., a player buys, plays, and comments on a game. Interactions that represent the players' experience are object of study of a very active research area called Player Modeling (PM). This paper presents* `SMMnet`*: the first dataset for PM that comes from a network of player-game interactions regarding the well-know Super Mario Maker (Nintendo, Kyoto, Japan).* `SMMnet` *presents a collection of over 880 thousand players that performed nearly 7 million interactions on over 115 thousand game levels. Moreover, we illustrate the diversity of the data with some statistical analyses and examples of studies.*

1. Introduction

The game universe is in constant ascendancy thanks to its popularity. For instance, there are streamers, which are famous players who develop online videos based on games [Gros et al. 2018]. Such ascendancy even created a new category of sports named eSports, which is a competition of professional players playing against each other [Yannakakis and Togelius 2018]. The game industry moves billions of dollars per year. According to Newzoo, a specialist company of game marketing, the estimated value for 2019 is U$ 150 billion.

With this expansion, the application of Artificial Intelligence (AI) in Games is growing with the need for more attractiveness and intelligence [Yannakakis and Togelius 2015]. Researchers focus on understanding the players' experience, *i.e.*, Player Modeling (PM), to study the players' behavior [Yannakakis et al. 2013, Aung et al. 2018]. In addition, from the perspective of game designers, players' behavior is one of the most important factors they must consider when designing the game systems [Lee et al. 2011, Yannakakis and Togelius 2018].

An ongoing research topic is PM in platforms of video games. A platform of video games is a virtual environment in which players interact with games [Eberhard et al. 2018]. The player-game interactions represent relationships of several types, *e.g.*, buy, play, and comment on a game. Each relationship can be represented by a social network, thus a platform of video games has a set of social networks; in this context, we coin the term Social Network of Games (SNG) for them.

For example, the well-known game Super Mario Maker (SMM) (Nintendo, Kyoto, Japan) is in fact a platform of video games of the Super Mario Bros series. In this platform, the players perform many types of interactions, *e.g.*, to play a game level (or simply game), give a "like", break a time record, comment on games created by other players, and elaborate his/her own levels to share online with the world.

Nowadays, there are several game datasets publicly available. Some of them focus on the industry games [Lee et al. 2011, Lin et al. 2017, Aung et al. 2018]; others refer to independent games [Lim and Harrell 2015, Karpouzis et al. 2015]. Nevertheless, these datasets are constrained in Player Modeling and game content exploration without any information from a Social Network of Games, thus making it difficult the study of Social Networks (SN) (*e.g.*, community detection, link prediction, ranking) on this context.

This paper presents the *first* very large and open access SNG dataset: the Super Mario Maker Network Dataset (or `SMMnet`, for short). It is publicly available for download [1]. The dataset provides information about over 115 thousand game levels and over 880 thousand players that performed nearly 7 million interactions of different types with the levels. `SMMnet` serves as a base for learning models, including, but not limited to, Player Modeling, Social Network Analysis, and general Data Mining, *e.g.*, prediction, and pattern discovery.

The rest of this paper is structured as in the following. First, we present formal definitions for Social Network of Games (Section 2). Then, we discuss the new dataset `SMMnet` (Section 3) and its applicability (Section 4). Finally, the last section concludes the paper (Section 5).

2. Social Networks of Games

A Social Network describes interactions and relationships of users in a digital environment [Barabási and Pósfai 2016]. In a network, users represent their relationships by links. There are many types of links, *e.g.*, links denoting social connections, similarity, behavioral interactions, actions, etc [Savić et al. 2019].

Links may also be present in a platform of video games, where players perform many types of interactions with games [Eberhard et al. 2018]. In these platforms, the players can buy, play, comment on a game, etc. Figure 1 illustrates two players p_1 and p_2 interacting (buy, play, comment, etc.) with three games c_1, c_2 and c_3. Each type of relationship form a social network; we coin the term Social Network of Games to refer to them in this paper.

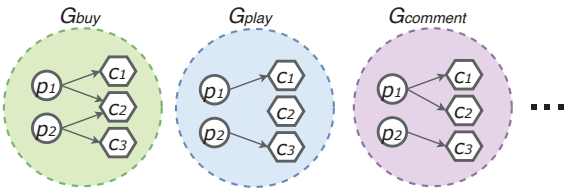


Figure 1. Examples of Social Network of Games.

In this sense, each relationship forms a social network represented by a directed bipartite graph $G = \{V, E\}$, with edges $e \in E$ of players $p \in V$ interacting with games $c \in V$, therefore $e = (p, c)$. Additionally, these networks change over time because new players/games arrive and new links appear. Therefore, since the graph changes as a function of time, it is a dynamic network [Westaby 2012]. Thereby, SNG has a series of snapshots of static graphs over time.

[1] `SMMnet`. *https://www.kaggle.com/leomauro/smmnet*

Note that a SNG is represented by a set of networks. Therefore, we can study each relationship network individually, given that some algorithms work only on homogeneous networks. Moreover, a SNG can also be represented by a complex network because the set of nodes may be connected by different and possibly overlapping types of relationships [Cherifi et al. 2017]. Also, each node can be represented by a complex object; *i.e.*, game features (*e.g.*, price, type of game, developer) and player features (*e.g.*, age, gender, time of playing).

2.1. Super Mario Maker

Super Mario Maker (SMM) is a platform of video games developed by Nintendo (Kyoto, Japan) for the consoles Nintendo Wii U[2] and Nintendo 3DS/2DS[3]. It was launched in September 2015. In this platform, a player can like and play game levels based on the Super Mario Bros series. Also, the player can create new game levels and share them online with the world. In this sense, SMM can be seen as a Social Network of Games.

In SMM, players present several types of relationships, as it is illustrated in Figure 2. A player can (1) create a game level and (2) play levels created by other players. If a player completes the challenge of the game level, he/she (3) "cleared" the level. The player can also be the (4) first to clear and/or beat the (5) time record of a level. Also, at any time, the player can (6) like a game level. In this sense, this Social Network of Games has six types of relationships.

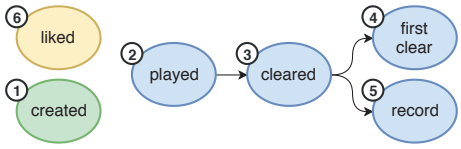


Figure 2. Types of player-game interactions on Super Mario Maker.

3. `SMMnet` Dataset

In this section, we describe the data collection in an automated fashion way. Furthermore, we describe the data, detail a schema for storing `SMMnet` into a Relational Database Management System (RDBMS), and conclude the section with some data analyses.

3.1. Data Collection

The information from Super Mario Maker is available in the website SMM Bookmark[4]. Thus, we elaborated a web crawler to collect the data from the website. A web crawler (or simply crawler) is a computer program that navigates on the world wide web in a systematic and automated way to search and/or collect data [Areekijseree et al. 2018]. To build our crawler, we searched for available Application Programming Interface (API) to retrieve information on the SMM Bookmark; and we found, in Node.js programming language, a set of APIs developed by independent programmers, *i.e.*, without connection with Nintendo.

[2]Nintendo Wii U. *http://bit.ly/nintendo-wii-u-smm* (accessed July 02, 2019).
[3]Nintendo 3DS/2DS. *http://bit.ly/nintendo-3ds-smm* (accessed July 02, 2019).
[4]SMM Bookmark. *https://supermariomakerbookmark.nintendo.net/* (accessed June 28, 2019).

However, the available APIs do not collect players' data, nor they capture the game changes (*e.g.*, new plays) over time. In this sense, we needed to adapt the most recent API, called `super-mario-maker-client`, to collect game changes. Also, we developed two new APIs: (I) `smm-maker-profile`, that collects players' data; and (II) `smm-course-search`, that searches for new game levels; both APIs are public available for download[5]. Figure 3 illustrates our crawler: `smm-course-search` searches for new game levels and stores their IDs into a Database (DB); `super-mario-maker-client` queries the game IDs in the DB and collects the games' data from SMM Bookmark; and, `smm-maker-profile` collects the players' data. It is important to notice that our crawler respects the access policies of the website SMM Bookmark (*i.e.*, *robots.txt*).

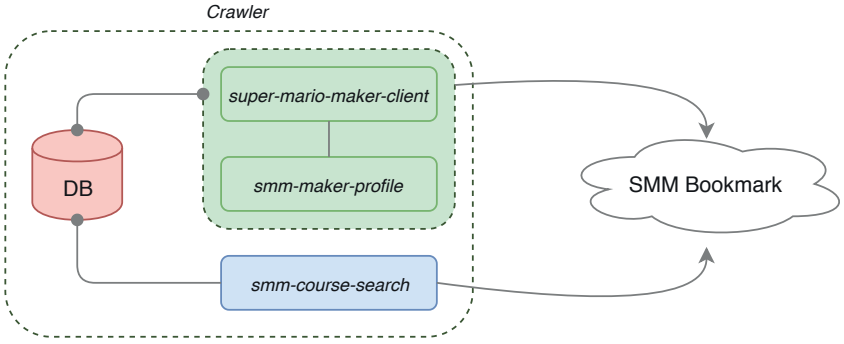


Figure 3. Crawler: Data collector structure.

However, the data collector suffered with bandwidth limits, as SMM Bookmark servers responded to a page request in around 2.7 seconds, making it impracticable to explore the whole network. Because of scalability considerations, we had to focus on a small set of games, maximizing the edge coverage over these groups of nodes aiming to preserve the community structure of the network sample [Areekijseree et al. 2018].

Therefore, we selected data from four nationalities to be collected: France (FR), Germany (DE), Canada (CA) and Brazil (BR). We selected FR, DE and CA countries for being the most active communities, right after United States and Japan, for which a huge number of levels were created per day, making it impractical to capture. We also selected BR to be a South America representative. Finally, the collection was performed during almost five months, from 16 Nov 2017 to 10 Apr 2018. The game changes from these countries were collected at every two hours.

3.2. Data Description

The `SMMnet` data is split into seven CSV (Comma-Separated Values) files: (1) COURSES.CSV, game level data; (2) COURSE-META.CSV, temporal changes on levels; (3) PLAYERS.CSV, players data; (4) PLAYS.CSV, plays; (5) CLEARS.CSV, clears; (6) LIKES.CSV, likes; and (7) RECORDS.CSV, time records over time. Figure 4 illustrates a schema with non-normalized tables to store the `SMMnet` into a Relational Database Management System (RDBMS). It is composed of seven tables, each one for one CSV file, that includes the levels, players, and the changes over time.

[5]npm. *https://www.npmjs.com/ leomaurodesenv* (accessed June 29, 2019).

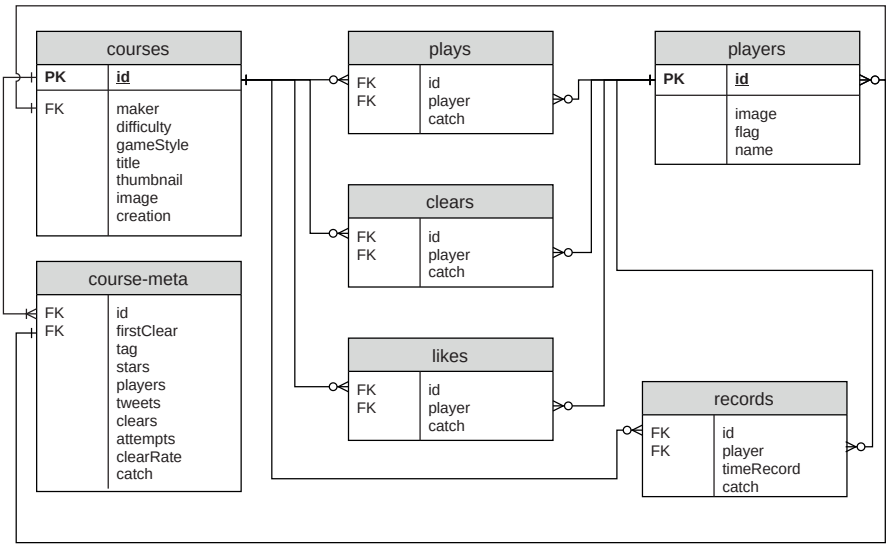


Figure 4. Schema for `SMMnet`.

COURSES.CSV presents static metadata from SMM levels. For example, level id (*string*, primary key), difficulty (*string*), game style (*string*), who created the game level - maker (*string*, foreign key), title (*string*), thumbnail link (*string*), image link (*string*), and creation date (*datetime*). Additionally, COURSE-META.CSV presents temporal changes on these levels, *i.e.*, tags (*string*), number of plays (*int*), number of tweets (*int*), clears (*int*), attempts (*int*), and likes (*int*) over time. All temporal tables have a "catch" field (*datetime*) that informs when the tuple was captured.

Meanwhile, combining the players and levels by links (*e.g.*, played, liked, cleared, and time records), forms the Social Network of Games. Thus, we need to correlate the tables COURSES and PLAYERS. PLAYERS.CSV presents the players' data, *i.e.*, id (*string*, primary key), player's image (*string*), name (*string*), and nationality - flag (*string*). While, we use auxiliary tables, PLAYS.CSV, CLEARS.CSV, LIKES.CSV, and RECORDS.CSV, to link the player and level tables over time, using a "catch" field.

Therefore, PLAYS.CSV, CLEARS.CSV, and LIKES.CSV only have three fields, *i.e.*, level id (*string*, foreign key), player (*string*, foreign key), and catch (*datetime*). Meanwhile, RECORDS.CSV has one more field, *i.e.*, time record in milliseconds (*int*).

Table 1 presents the number of tuples of each table. This dataset presents 115 thousand levels that 880 thousand players around the world played 3.94 million times, cleared 2.05 million times with 117 thousand time records, and liked 619 thousand times. In summary, the `SMMnet` is split into seven files that can be stored into a RDBMS, such as PostgreSQL, Oracle, and MySQL. The next section presents more dataset characteristics.

3.3. Dataset Characteristics

Figures 5 and 6 illustrate the proportion of the game styles and game difficulties of levels, respectively. Considering Figure 5 and Table 2, 51.7% of the levels follow the game style Super Mario Bros U, 21.2% follow Super Mario Bros, 19.0% follow Super Mario World, and 8.1% follow Super Mario Bros 3. Note, there is a high preference for the most recent game style (*i.e.*, Super Mario Bros U), followed by the oldest game style (*i.e.*, Super Mario

Table 1. Quantity of tuples in each table.

Table	Data
COURSES	115k
COURSE-META	292k
PLAYERS	884k
PLAYS	3,940k
CLEARS	2,050k
LIKES	619k
RECORDS	117k

Bros). In this sense, we presume that the most recent style prevails through the new visual and gameplay characteristics, while the oldest style stands out due to the memory of the most traditional style of the Super Mario Bros series.

Figure 6 and Table 3 show the percentages of courses according to their difficulties. In order of difficulty, 25.8% of the courses are considered easy, 44.6% are normal, 24.1% are expert, and 5.4% are super expert. To the best of our knowledge, there is no rule to define the game difficulty; SMM platform establishes the difficulty of a level automatically.

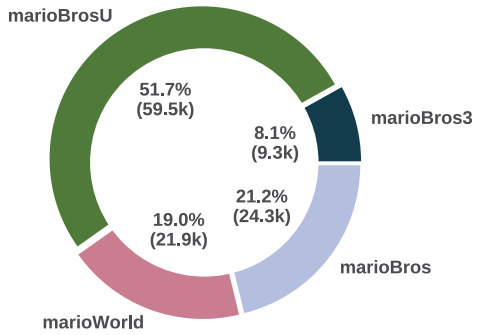


Figure 5. Game levels styles.

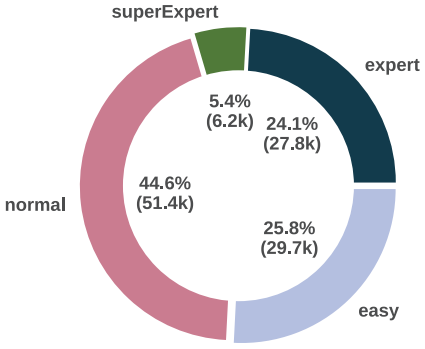


Figure 6. Game levels difficulties.

Table 2. Game levels styles.

Style	Data
marioBrosU	59.5k (51.7%)
marioBros	24.3k (21.2%)
marioWorld	21.9k (19.0%)
marioBros3	9.3k (08.1%)

Table 3. Game levels difficulties.

Difficulty	Data
easy	29.7k (25.8%)
normal	51.4k (44.6%)
expert	27.8k (24.1%)
superExpert	6.2k (05.4%)

The total number of interactions (*i.e.*, plays, clears, records and likes) made by players is 6,729,000. Among them, 3,941,378 refer to interactions of the type play, while 2,051,809 are clears, 117,126 are time records, and 618,687 are likes. Figure 7 shows the types of interactions on the top-100 game levels with the highest number of interactions. As we can see, many plays, likes and clears exist, but few time records occur because a level usually has few broken records.

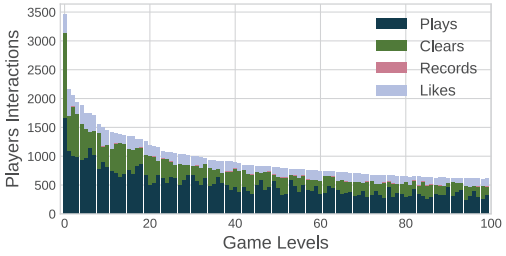


Figure 7. Dataset: Sum of the players interactions by level for the top-100 most popular game levels.

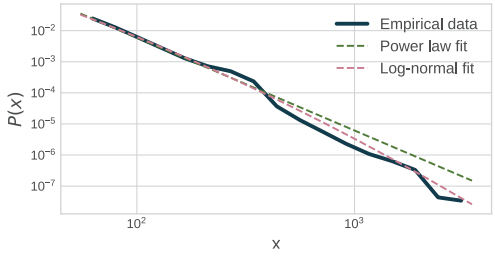
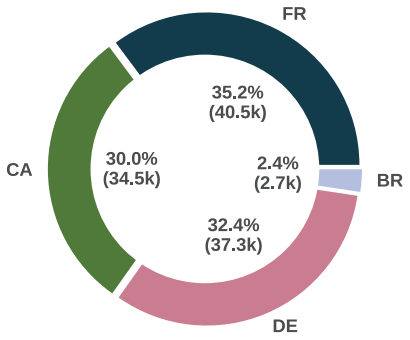


Figure 8. Probability Density Function for the top-100 most popular game levels; with power-law and log-normal estimates.

The interaction data in Figure 7 presents a few levels with high interaction and a fast decrease. This behavior is similar to a power-law, which is very common in Social Networks, including few popular and many unpopular objects [Newman 2005]. In this dataset, only 0.13% (151) levels received more than 500 interactions and the majority 99.87% (114,881) has interactions ranging from 0 to 500. Figure 8 reports a Probability Density Function plot for the top-100 most popular game levels. As it can be seen, it exhibits a similar behavior of a power-law [Clauset et al. 2009] and log-normal distribution estimates [Mitzenmacher 2004].

In `SMMnet`, there are over 880 thousand players that performed 7 million interactions on over 115 thousand levels. Besides, as mentioned before, we selected four nationalities, collecting 41 thousand levels from FR, 37 thousand from DE, 34 thousand from CA, and three thousand from BR. Figure 9 and Table 4 summarize this information. This is a substantial amount of data to infer knowledge about the players and games in at least four different countries. Additionally, there are no missing values.

Figure 9. Game levels by country.



Country	Data
France (FR)	40.5k (35.2%)
Germany (DE)	37.3k (32.4%)
Canada (CA)	34.5k (30.0%)
Brazil (BR)	2.7k (02.4%)

Table 4. Game levels by country.

4. Applicability

The `SMMnet` dataset can be used in several applications, *e.g.*, Social Network and Artificial Intelligence in Games studies. Next, we present some studies for motivating potential users of the dataset to find other creative uses.

4.1. Social Networks

It is possible form at least three types of graphs in this dataset, *i.e.*, static graphs, dynamic graphs, and complex networks. Exploring a static graph requires to observe the social network at a specific timestamp (snapshot), *e.g.*, a social network of likes from the last day. In a dynamic graph, the social networks change over time. Thus, it is necessary to elaborate temporal graphs at each given time interval, *e.g.*, a social network of likes for each day. Meanwhile, in a complex network, it is necessary to take into account the multiple relationship types and/or complex objects, *e.g.*, a social network with plays and clears links.

In this sense, we emphasize that this dataset can be used in different scenarios. `SMMnet` can be explored in many social network studies, including, but not limited to: (1) community detection, to identify communities of players (*e.g.*, similar players); (2) link prediction, *e.g.*, infer what games a player will play on a network of plays; and (3) ranking, *e.g.*, sort the popular games, or influential players.

4.2. Artificial Intelligence in Games

Researchers attest that companies invest in production of new games with graphics and interactive qualities by using artificial intelligence techniques [Lucas 2009, Yannakakis 2012]. Besides, games are advantageous test-bed to algorithms because they offer a simulated environment in which many factors are reacting simultaneously.

This dataset offers many artificial intelligence applications. We highlight some possibilities: (1) Data Mining, supervised or unsupervised learning (*e.g.*, clustering to identify similar games' characteristics); and (2) Player Modeling, extract characteristics from the players' activities in the social networks.

4.2.1. Detecting Influential Players

`SMMnet` has already been used in some studies that illustrate its usefulness in Player Modeling. Researchers used this Social Network of Games dataset to detect the game influencers (or simply influencers), that is, players with high influence in creating new trends by publishing online contents (*e.g.*, videos, blogs, forums) [Moraes and Cordeiro 2019].

Other players follow the influencers looking for entertainment and credible information about the games. Consequently, game companies invest in influencers to perform marketing for their products. However, it is not a trivial task to detect game influencers among thousands of players. Moraes and Cordeiro (2019) proposed a framework to extract temporal aspects of the players' actions, and then detect the influencers by performing a classification analysis.

Figure 10 illustrates the framework proposed by Moraes and Cordeiro (2019), which is split into three steps: network modeling; player modeling; and classification analysis. In the network modeling process, they model the social network of developments (G_{dev}) and likes (G_{like}); both graphs are dynamic and directed bipartite. G_{dev} represents a network of player creations, *i.e.*, the player who created each game level.

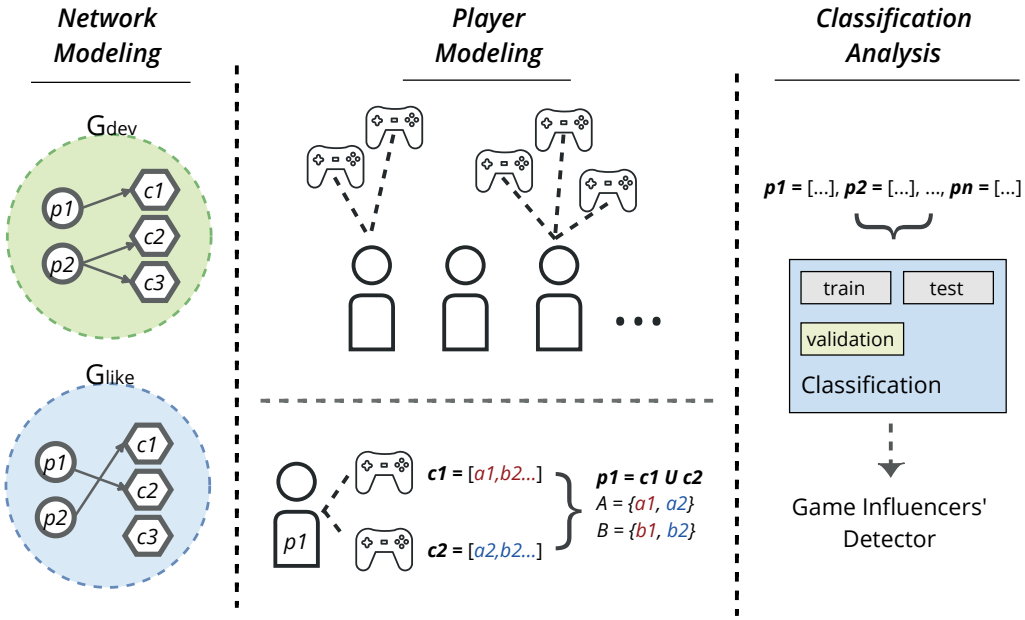


Figure 10. Framework for Detecting Game Influencers.

Meanwhile, G_{like} represents a network with game levels liked by players; it is a network of the levels liked over time.

In the player modeling step, the researchers extracted a set of game levels created by each player, using the G_{dev} network. For each game level it was modeled a data stream of likes over time, using the G_{like} network. For each data stream of likes, it was extracted a series of features, and then combined to represent the player's features of its creator. Therefore, a player was represented by a combination of his/her games' features.

In classification analysis, they evaluated 28 classification algorithms using the players' features. The best classifier (Logistic Regression) reached high accuracy (87.1%) and f1-score (85.7%) to detect the game influencers. Note that player labels (*i.e.*, ground truth) were created manually by observing their activity on popular gaming sites. The authors also executed a validation, which demonstrated that the proposed framework automatically detects influencers with high precision even when using data from distinct countries for testing and training.

5. Conclusion

In this paper, we presented `SMMnet`: the *first* dataset for Social Networks of Games (SNG). SNG is a set of social networks in which the players perform many types of interactions with games, *e.g.*, buy, play, and like a game. In this sense, an SNG can be represented by a set of social networks or a complex network. Also, these networks can change over time; therefore, they have dynamic graphs as function of time.

`SMMnet` was extracted from the well-known game Super Mario Maker (Nintendo, Kyoto, Japan). This dataset presents a collection of over 880 thousand players that performed nearly 7 million interactions on over 115 thousand levels. Besides, we present a schema to store this dataset into a Relational Database Management System. Finally, we highlighted the applicability of this dataset in the research fields of Social Networks

and Artificial Intelligence in Games. However, its limitation is not having internal data of the game levels, only metadata from social networks. In addition to the presented sample studies, we believe that researchers and game designers will find further creative proposals for this dataset.

Acknowledgements

This research was supported by the Brazilian National Council for Scientific and Technological Development (CNPq); Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) [grant 001]; Sao Paulo Research Foundation (FAPESP) [grants 2018/05714-5 and 2016/17078-0]; and AWS Cloud Credits for Research.

References

Areekijseree, K., Laishram, R., and Soundarajan, S. (2018). Guidelines for online network crawling: A study of data collection approaches and network properties. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '18, pages 57–66, New York, NY, USA. ACM.

Aung, M., Bonometti, V., Drachen, A., Cowling, P., Kokkinakis, A. V., Yoder, C., and Wade, A. (2018). Predicting skill learning in a large, longitudinal moba dataset. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–7.

Barabási, A.-L. and Pósfai, M. (2016). *Network science*. Cambridge university press, Cambridge, USA.

Cherifi, C., Cherifi, H., Karsai, M., and Musolesi, M. (2017). *Complex Networks & Their Applications VI: Proceedings of Complex Networks 2017 (The Sixth International Conference on Complex Networks and Their Applications)*, volume 689 of *Studies in Computational Intelligence*. Springer.

Clauset, A., Shalizi, C., and Newman, M. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703.

Eberhard, L., Kasper, P., Koncar, P., and Gütl, C. (2018). Investigating helpfulness of video game reviews on the steam platform. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 43–50.

Gros, D., Hackenholt, A., Zawadzki, P., and Wanner, B. (2018). Interactions of twitch users and their usage behavior. In Meiselwitz, G., editor, *Social Computing and Social Media. Technologies and Analytics*, pages 201–213, Cham. Springer International Publishing.

Karpouzis, K., Yannakakis, G. N., Shaker, N., and Asteriadis, S. (2015). The platformer experience dataset. In *2015 International Conference on Affective Computing and Intelligent Interaction*, pages 712–718, USA. ACII.

Lee, Y.-T., Chen, K.-T., Cheng, Y.-M., and Lei, C.-L. (2011). World of warcraft avatar history dataset. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, MMSys '11, pages 123–128, New York, NY, USA. ACM.

Lim, C.-U. and Harrell, D. F. (2015). Comparing clustering approaches for modeling players' values through avatar construction. *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Lin, Z., Gehring, J., Khalidov, V., and Synnaeve, G. (2017). Stardata: A starcraft ai research dataset. *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Lucas, S. M. (2009). Computational intelligence and ai in games: A new ieee transactions. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(1):1–3.

Mitzenmacher, M. (2004). A brief history of generative models for power law and log-normal distributions. *Internet Mathematics*, 1(2):226–251.

Moraes, L. M. P. and Cordeiro, R. L. F. (2019). Detecting influencers in very large social networks of games. In *Proceedings of the 21st International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 93–103, Crete, Greece. INSTICC, SciTePress.

Newman, M. (2005). Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351.

Savić, M., Ivanović, M., and Jain, L. C. (2019). *Introduction to Complex Networks*, pages 3–16. Springer International Publishing, Cham.

Westaby, J. D. (2012). *Dynamic network theory: How social networks influence goal pursuit.* American Psychological Association.

Yannakakis, G. N. (2012). Game ai revisited. In *Proceedings of the 9th Conference on Computing Frontiers*, CF '12, pages 285–292, New York, NY, USA. ACM.

Yannakakis, G. N., Spronck, P., Loiacono, D., and André, E. (2013). Player modeling. In Lucas, S. M., Mateas, M., Preuss, M., Spronck, P., and Togelius, J., editors, *Artificial and Computational Intelligence in Games*, volume 6 of *Dagstuhl Follow-Ups*, pages 45–59. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany.

Yannakakis, G. N. and Togelius, J. (2015). A panorama of artificial and computational intelligence in games. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(4):317–335.

Yannakakis, G. N. and Togelius, J. (2018). *Artificial Intelligence and Games*. Springer International Publishing, Cham.

SoccerNews2018: a dataset of statistics and news of the 2018 Brazilian Soccer Championship

Júlio César Machado Álvares[1], **Marcos Roberto Ribeiro**[1]

[1] Instituto Federal de Minas Gerais (IFMG), Bambuí, Brazil

juliocmalvares07@gmail.com, marcos.ribeiro@ifmg.edu.br

Abstract. *Soccer is a worldwide known sport and the most famous sport in Brazil. This fact has always caused a lot of speculation regarding important championships and matches, club relegation and other events involving the sport. Thus, also in science, several research works try to predict the outcome of soccer matches. However, the vast majority of research efforts use only statistical data to predict results. The construction of new predictive models for soccer match results depends on the existence of databases containing more information than pure statistical data. This paper presents a database of the 2018 Brazilian Soccer Championship containing statistical data and news about the teams. Here, we describe the developed tools for data extraction and the structure of the dataset. We also highlight the main applications for our dataset.*

1. Introduction

Soccer is a worldwide known sport and the most famous sport in Brazil [Araújo et al. 2015]. This fact has always caused a lot of speculation regarding important championships and matches, club relegation and other events involving the sport. Thus, also in science, several research works try to predict the outcome of soccer matches [Araújo et al. 2015, da Silva 2018, Passos and Kronbauer 2018, Filho et al. 2017, Rue and Salvesen 2000]. However, the vast majority of research efforts use only statistical data to predict results.

The construction of new predictive models for soccer matches results depends on the existence of databases containing more information than pure statistical data. As examples of additional information, we can highlight news, user comments, attendance and, gameplays. As far as we know, no datasets are having more information than match statistics[1]. In this way, the main goal of this work is to present a database of the 2018 Brazilian Soccer Championship containing statistical data and news about the teams. This data can be useful for data mining tasks such as classification, pattern recognition, and even anomaly detection.

Importantly, there are no publicly available datasets that match the one presented in this paper. Most publicly available datasets are from English championships, or from the Europe region. So, it is expected to obtain Brazilian databases that have good quality for studies.

This paper is organized as follow. First, Section 2 presents the related works. Section 3 explains how the data extraction is processed. Next, Section 4 describes the dataset. In the following, Section 5 highlights the research opportunities. Section 6 presents some limitations of the dataset. Finally, Section 7 concludes this paper.

[1] Int the sports context, the match statistics are just the summary of match events.

2. Related Works

[Carpita et al. 2019] in their work explore a public football database available at Kaggle[2], which contains information about teams from 11 countries in Europe and its respective championships. The database, as presented in this paper, contains information about matches, with the same fields presented here, such as goal kicks, cards, ball possession, among others. The authors discuss and present tests on famous prediction algorithms such as KNN, Random Forest, Neural Networks and a Binomial Logistic Regression. However, there are some differences, positive and negative, regarding our database and that presented by the authors.

First, as a positive point, the database that the authors explore contains player and team-related attributes such as field formations, player performance information, and others, as well as 8-year data (2008 to 2016). On the downside, the database only contains information referred to as statistics, and does not contain any other information, such as team news, for the database presented here.

In other work, [Dubitzky et al. 2019] presents a fairly extensive database of international football matches that addresses a multitude of teams from around the world. However, like the article previously cited in this Section, this paper has its advantages and disadvantages. Firstly, as an advantage, the database presented contains information about various teams from around the world, as well as various editions of various championships, but as a negative point, the amount of information about each game is extremely small. The database delivers only 2 relevant statistical data on each match, which are the goals of each team and the goal balance.

[Tunaru and Viney 2010] in their work, they have developed a framework for predicting the market value of football team players, based on the statistics of games they played, as well as other measures such as player personal performance, injuries and others. The database the authors use is maintained by OPTA Sports data[3], a London-based sports data company. However, as with the other papers presented in this Section, the [Tunaru and Viney 2010] job database does not present complementary information to the teams, only statistics. By observing the works exposed here, it is understood that there are no known public databases that contain extra information to statistics in the field of football, as presented in this article.

3. Data Extraction

The first step to building the SoccerNews2018 dataset was to get the list of teams that competed for the 2018 Brazilian Championship. Based on this list, we extracted the team news from the GloboEsporte website[4]. This portal contains news about the Brazilian Championship teams up to 5 years ago, as well as news on a wide variety of sports. First, we delimited the news dates beginning with the news up to one week before the championship beginning (Apr/14) and ending with the last day of matches (Dec/02).

The portal groups the information by teams, but does not have an RSS news feed. Therefore, we analyzed the source code of the portal and developed a web crawler to get

[2] https://www.kaggle.com/
[3] https://www.optasports.com/
[4] https://globoesporte.globo.com/

and store all the news links of a certain team. Then we built a second crawler to access the news and extract their data. Both crawler systems were developed in the Python language, with the help of the Scrapy library[5].

Regarding statistics, we started by getting the listing of all the championship matches on Wikipedia[6]. For each match, we used the Veja Portal[7] to extract statistics information. This portal has the statistics of all matches through a custom URL and also other data that can be used in the future as the play-by-play events. We also developed a crawler to automate the extraction of the statistics. Due to the dynamic components of the Veja to generate the pages, we used the Selenium library[8]. This library is useful to get the content of the pages exactly as shown in web browsers.

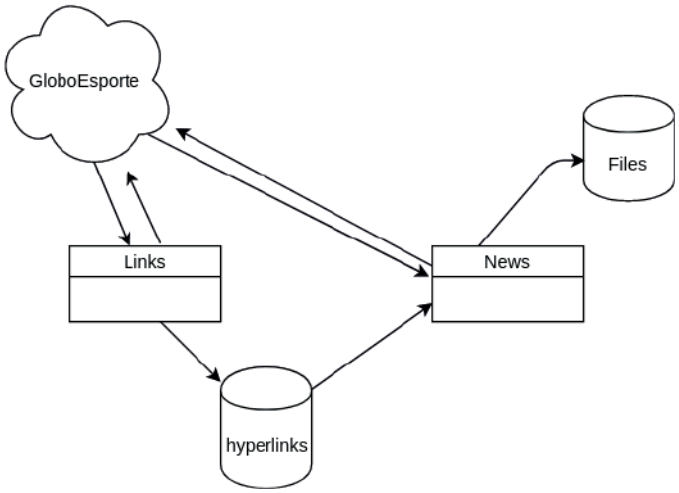


Figure 1. Crawler for news extraction

Figure 1 shows the working scheme of the news crawler. This crawler is composed of the `Links` and `News` entities. The `Links` entity initiates the extraction of links in the page `https://globoesporte.globo.com/futebol/times/$TEAM$/index/feed/pagina-$N$.ghtml` where `$TEAM$` is a team and `$N$` starts at 1. The system increments iteratively the value of N until pages having news before the defined deadline be found. By accessing the pages, the crawler collects the news links and stores them in the `hyperlinks` file. The entity `News`, in turn, visits each of the links to extract the news content. The data of interest as text, title and date of the news are stored in `Files`. This entity consists of a set of files, one for each news, where all news about the teams are stored.

Figure 2 shows the working scheme for statistics crawler. In this crawler, the `hyperlinks` file already has the links of all the championship matches. The statistics can be obtained through the URL `https://veja.abril.com.br/placar/campeonato-brasileiro/$HOME$-e-$VISITOR$-$DATE$/` where `$HOME$` is the home team, `$VISITOR$` is the visiting team and `$DATE$` is the match date. The `Crawler` entity visits all these links, using the `Chrome` browser emulator and stores the

[5] https://scrapy.org/

[6] https://pt.wikipedia.org/wiki/Campeonato_Brasileiro_de_Futebol_de_2018_-_Série_A

[7] https://veja.abril.com.br/

[8] https://selenium-python.readthedocs.io/

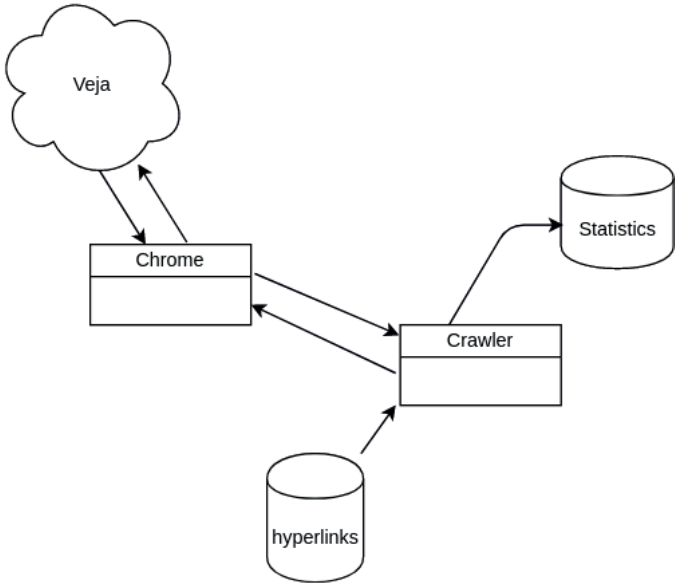


Figure 2. Crawlers for statistics extraction

data obtained in the `statistics` file.

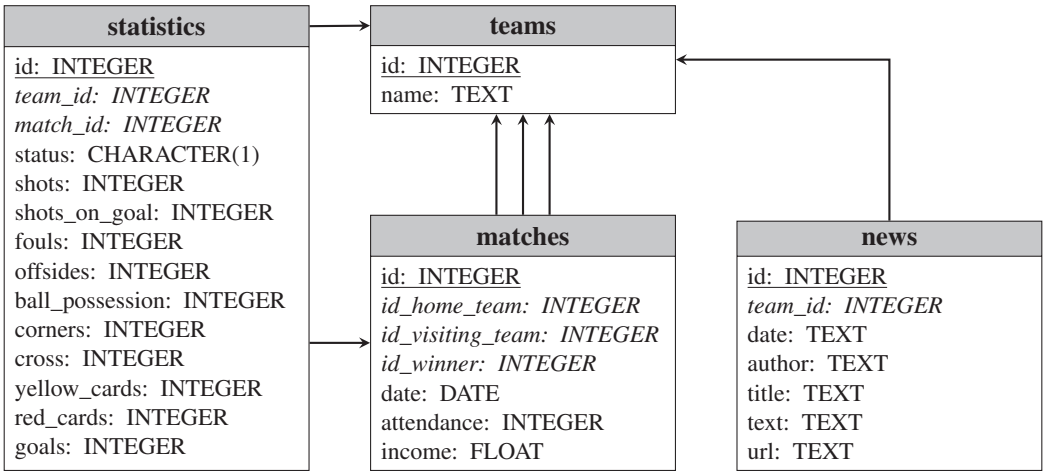


Figure 3. Logic schema of the dataset

4. The Dataset

Figure 3 presents the logic schema to the SoccerNews2018 dataset. The dataset is composed of the following relations:

`teams`: All participant teams of the Brazilian Soccer Championship 2018;

`matches`: Matches of the championship;

`statistics`: Statistics of the matches;

`news`: News about the teams.

The relation `teams` has only the attributes `id` (team identifier) and `name` (team name). Table 1 shows the attributes of relation `news`. The content of `news` can be filtered by team or date. By using the team identifier, we can make various interesting queries such as the relationship between news and matches.

Table 1. Attributes of relation `news`

Attribute	Description
id	Identifier
team_id	Team identifier
date	Publication date
author	Author of the new
title	Title of the new
text	Full text of the new
url	Original URL

Table 2. Attributes of relation `matches`

Attribute	Description
id	Identifier
date	Match date
attendance	The audience present
income	Income in Brazilian Real
id_home_team	Identifier of home team
id_visiting_team	Identifier of visiting team
id_winner	Identifier of winner

Tables 2 and 3 present the attributes of relations `matches` and `statistics`, respectively. The relation `matches` has three relationships to the relation `teams` to determine the home team, the visiting team and the match-winner. The relation `statistics` is linked to relations `teams` and `matches`. So, it is possible to obtain the statistics of a team in each match.

Table 3. Attributes of relation `statistics`

Attribute	Description
id	Identifier
team_id	Team identifier
match_id	Match identifier
status	Home or visitor
shots	All shots
shots_on_goal	Shots on goal direction
fouls	Committed fouls
offsides	Number of off-sides
ball_possesion	Ball possesion
corners	Number of corners
cross	Crosses do goal area
yellow_cards	Yellow Cards
red_cards	Red Cards

The full dataset and the crawlers are available for download in a Github repository[9]. The dataset can be downloaded in JSON format. In addition, the repository contains data and crawlers, as well as some utilities for data formatting and conversion to other formats. Table 4 presents the number of tuples for each relation of the dataset.

5. Research Opportunities

The dataset SoccerNews2018 is useful for a multitude of studies. This section describes a non-exhaustive list of research fields that can use our dataset.

[9] `https://soccerpredict.github.io/TeamNews/`

Table 4. Data statistics

Relation	Tuples
Teams	20
Matches	374
Statistics	748
News	16.553

Data Mining. In the Data Mining field, our dataset can be used to develop new techniques for predicting match results since the vast majority of research works use only statistical data for this task [Snyder 2013, da Silva 2018, Rue and Salvesen 2000, Passos and Kronbauer 2018, Filho et al. 2017, Araújo et al. 2015]. In addition, the dataset can be explored by pattern mining, outlier detection and sentiment analysis [Jai-Andaloussi et al. 2015]. It would be possible to use sentiment analysis approaches in the news content and check if they have impact on the match results.

Sport Analysis. In this field, our dataset can be used by sports experts in order to have useful information for their team, using techniques such as data mining or data visualization. The expert can use statistical data to set up strategies for the team and use the news for marketing activities [Rein and Memmert 2016, Miljković et al. 2010].

6. Limitations

The SoccerNews 2018 dataset used only statistics provided by the Veja portal. Some less popular stats are not available on the portal. Therefore, you could enrich the dataset with more statistical information using other data sources.

Another limitation concerns the news. They were obtained only in the GloboSport portal. This portal was chosen for presenting an interesting amount of news from all the teams of the Brazilian championship. So, an improvement in the dataset would get more news from other portals. In addition, user comments on each news item can also be added to the database.

Still regarding the previous limitation, it can be understood that, since it is only data from the year 2018, this is a limitation. However, this paper proposes only the database from the year 2018. Initially, it is possible to obtain data from other years of the championship using our system. The only problem, as it is a crawler, is that the portal pages are not available.

7. Conclusion

In this article, we presented the SoccerNews2018 dataset containing news and statistics about the 2018 Brazilian Soccer Championship. The dataset was built using two information retrieval systems. Both the dataset and the developed tools have been made available under a free license in a GitHub repository. The dataset can be used in various types of research, mainly, in the fields of data mining and sports analysis. The authors are currently working on a research project intended at analyzing the impact of team news on match results. In addition to this project, the authors are also working on improvements to the dataset, bringing more features and more championship editions. From this analysis, existing prediction methods can be improved based on news content.

Acknowledgments. The authors thanks the Research Agencies CNPq, CAPES and FAPEMIG for supporting this work.

References

Araújo, C., Tavares, L., Alvares, L., Neto, F., and Suzuki, A. (2015). Modelagem estatística para previsão de jogos de futebol: Uma aplicação no campeonato brasileiro de 2014. *Revista de Estatística da Universidade de Ouro Preto*.

Carpita, M., Ciavolino, E., and Pasca, P. (2019). Exploring and modelling team performances of the kaggle european soccer database. *Statistical Modelling*, pages 74 – 101.

da Silva, B. M. (2018). Regressão linear múltipla aplicada ao futebol. *Revista Brasileira de Futebol e Futsal*, 10:262–270.

Dubitzky, W., Lopes, P., Davis, J., and Berrar, D. (2019). The open international soccer database for machine learning. *Machine Learning*, 108(1):9 – 28.

Filho, C., Suzuki, A., Louzada, F., Saraiva, E., and Salasar, L. (2017). Uma abordagem bayesiana para previsão de resultados de jogos de futebol: Uma aplicação ao campeonato inglês. *Revista Brasileira de Biometria*.

Jai-Andaloussi, S., Mourabit, I. E., Madrane, N., Chaouni, S. B., and Sekkaki, A. (2015). Soccer events summarization by using sentiment analysis. *International Conference on Cumputational Science and Computational Intelligence*, pages 398–403.

Miljković, D., Gajić, L., Kovačević, A., and Konjović, Z. (2010). The use of data mining for basketball matches outcomes prediction. In *IEEE 8th International Symposium on Intelligent Systems and Informatics*, pages 309–312.

Passos, V. and Kronbauer, D. P. (2018). Uma abordagem estatística em aprendizagem de máquina para previsões em campeonatos de futebol. *Anais SULCOMP*.

Rein, R. and Memmert, D. (2016). Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. *SpringerPlus*.

Rue, H. and Salvesen, O. (2000). Prediction and retrospective analysis of soccer matches in a league. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49(3):399–418.

Snyder, J. A. L. (2013). What actually wins soccer matches: Prediction of the 2011-2012 premier league for fun and profit. *Master's thesis, Princeton University, NJ*.

Tunaru, R. S. and Viney, H. P. (2010). Valuations of soccer players from statistical performance data. *Journal of Quantitative Analysis in Sports*, 6(2).

REALIZATION

EXECUTION

SUPPORT

SILVER SPONSOR

ACADEMIC SUPPORT