August 25 and 26 • Rio de Janeiro - Brazil

33rd Brazilian Symposium on DATABASES

Proceedings Companion

Realização
SBC Sociedade Brasileira de Computação

Organização
IME

Apoio
CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico
CAPES

Patrocínio
Google

Apoio Institucional
PUC RIO

XXXIII BRAZILIAN SYMPOSIUM ON DATABASES (SBBD 2018)

August 25-26, 2018
Rio de Janeiro, RJ, Brazil

PROCEEDINGS Sociedade Brasileira de Computação (SBC)

PROMOTION
Sociedade Brasileira de Computação (SBC)

ORGANIZATION
Instituto Militar de Engenharia (IME)

SUPPORT
Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)
Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)

SPONSOR
Google

ACADEMIC SUPPORT
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

005.74 Brazilian Symposium on Databases (SBBD 2018) (25.: 2018: Rio de Janeiro, RJ)

B827 Proceedings of 33nd Brazilan Symposium on Databases (SBBD 2018): August 25-26, 2018 – Rio de Janeiro, RJ, Brazil; organizadores: Maristela Holanda; José Maria Monteiro – Rio de Janeiro: SBC, 2018.

ISSN: 2016-5170
volume 02
123p.
Modo de acesso: http://sbbd.org.br/2018/

1. Computação - Congressos. 2. Bases de Dados – Congressos. l. Holanda, Maristela. II. Monteiro, José Maria. III. Sociedade Brasileira de Computação. IV. Título.

Message from the Local Organization Committee Chair

Welcome to the 33rd Brazilian Symposium on Databases! 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 33rd edition of the symposium (SBBD 2018) was held in Rio de Janeiro, RJ, from August 25th to 26th, 2018. The local organization was performed by Instituto Militar de Engenharia (IME).

This year, for the first time, SBBD was held right before and at the same location of the 44thInternational Conference on Very Large Databases Conference (VLDB 2018), one of the main international conferences in the database area. This was a great opportunity for gathering national and international database researchers.

The SBBD 2018 program offered a variety of activities, suited for an audience ranging from undergraduate to Ph.D. students, database professionals, practitioners and researchers. The Program included: 4 invited talks and 2 tutorials, presented by distinguished speakers from Brazil, USA and Switzerland; 6 technical sessions; demos and applications session; posters sessions; thesis and dissertations workshop.

The excellence of SBBD 2018 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 2018 edition. We thank the symposium chairs and our colleagues of the local organization committee who donated their precious time to make SBBD 2018 a reality. We specially thank IME and its Post-graduation Programs (PGSC and PGED), 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 2018. 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 had a great time hosting SBBD 2018 in Rio de Janeiro!

Maria Claudia Reis Cavalcanti (IME)
Local Organization Chair – SBBD 2018

Editorial for the Demos and Applications Track

The Brazilian Symposium on Databases (SBBD) is the largest venue in Latin America for presenting research results in the database domain. In its 33nd edition, SBBD will be held in Rio de Janeiro, from August 25th to 26th, 2018.

The Demonstrations and Applications Session is organized since 2004 within SBBD. The Demostrations Session 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 3 reviewers selected from a committee of 24 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 2018 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 Rio de Janeiro!

Maristela Holanda (UnB)
Program Chair – SBBD 2018 – Demos and Applications

Editorial for the Workshop on Thesis and Dissertations in Databases

The Workshop on Thesis and Dissertations in Databases (WTDBD) is a traditional event co-located with the Brazilian Symposium on Databases (SBBD), the largest venue in Latin America for presenting research results in the database domain. This year, the event takes place in Rio de Janeiro, gathering professors and graduate students from different Universities in Brazil to present and discuss their most recent results.

The WTDBD is an excellent opportunity to receive feedback upon on-going graduate work from experienced researchers. All submitted papers received, at least, three reviews. Additionally, during the Workshop, students of selected papers have the opportunity to present their work and to receive technical and scientific comments, as well as experimenting the challenge of presenting their research to an external committee. In this edition, we have eleven accepted works (seven masters and four doctorate works) from many different universities in Brazil, selected from a committee of 23 researchers.

The 2018 WTDBD 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 for their high quality evaluations. Their insightful comments will probably have positive impact in the development of the different research initiatives presented in the WTDBD. Finally, the WTDBD coordinator would like to thank the SBBD 2018 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 works.

José Maria Monteiro (UFC)
Program Chair – SBBD 2018 – Workshop on Thesis and Dissertations in Databases

Demos and Applications – Technical Committee

Program Chair

Maristela Holanda (UnB)

Program Committee

Angelo Brayner (UFC)
Cristina Ciferri (USP)
Damires Souza (IFPB)
Daniel de Oliveira (UFF)
Daniel Kaster (UEL)
Eduardo Bezerra (CEFET/RJ)
Eduardo de Almeida (UFPR)
Eduardo Ogasawara (CEFET/RJ)
Fabio Porto (LNCC)
Flávio R. C. Sousa (UFC)
Humberto Razente (UFU)
Jonas Dias (Dell EMC)
José Maria Monteiro (UFC)
José de Aguiar Moraes Filho (UNIFOR)
Kary Ocaña (LNCC)
Leonardo Azevedo (IBM Research Brazil)
Luiz André Paes Leme (UFF)
Marcela Ribeiro (UFSCar)
Marcio Oikawa (UFABC)
Marcio Victorino (UnB)
Mirella Moro (UFMG)
Renata Galante (UFRGS)
Rodrigo Monteiro (UFF)
Ronaldo Mello (UFSC)

Workshop on Thesis and Dissertations in Databases – Technical Committee

Program Chair

José Maria Monteiro (UFC)

Program Committee

Altigran Soares da Silva (UFAM)
Ana Carolina Almeida (UERJ)
Angelo Brayner (UFC)
Carina F. Dorneles (UFSC)
Carlos Eduardo Pires (UFCG)
Daniel de Oliveira (UFF)
Daniel Kaster (UEL)
Eduardo de Almeida (UFPR)
Flávio R. C. Sousa (UFC)
Francisco Nauber Bernardo Gois (UFC)
Jonice Oliveira (UFRJ)
José Antonio Macêdo (UFC)
José de Aguiar Moraes Filho (UNIFOR)
Karin Becker (UFRGS)
Leonardo Moreira (UFC)
Luciano Barbosa (UFPE)
Mirella Moro (UFMG)
Renata Galante (UFRGS)
Renato Fileto (UFSC)
Ronaldo Mello (UFSC)
Sergio Lifschitz (PUC-Rio)
Valéria C. Times (UFPE)
Vania Vidal (UFC)

Table of Contents

Demos and Applications

Lathe: light-Weight Keyword Query Processing over Multiple Databases
Pericles de Oliveira, Altigran da Silva, Edleno Moura, Gilberto Santos 1

VP-Viewer: keeping Track of Your Query from a Vantage Point
Daniel L. Jasbick, Thaylon Guedes, Rodolfo A. Oliveira, Lúcio F.D. Santos, Daniel de Oliveira, Marcos V.N. Bedo 5

Metamorfose: a Data Transformation Framework Based on Apache Spark
Evandro Miguel Kuszera, Leticia M. Peres, Marcos Didonet Del Fabro 11

Análise Online de Dados de Proveniência e de Domínio de Aplicações Spark com SAMbA
Thaylon Guedes, Vítor Silva, Marcos V.N. Bedo, Marta Mattoso, Daniel de Oliveira 17

ImgDW Generator: a Tool for Generating Data for Medical Image Data Warehouses
Guilherme Muzzi da Rocha, Cristina Dutra de Aguiar Ciferri 23

Outer-Tuning: sintonia Fina Automática Baseada em Ontologia
Ana Carolina Almeida, Edward Hermann Haeusler, Sérgio Lifschitz, Rafael Pereira de Oliveira, Daniel Schwabe 29

Anelim: uma Ferramenta de Geração Automática de Dados para Banco de Dados Relacional em Ambientes de Testes
Angelo Brayner, F. Ronald Araújo B., José Maria Monteiro 35

MobileECG: uma Ferramenta para Publicação e Integração de Dados de Sinais ECG
Tibet Teixeira, Francisco San Diego Castilho, Daniel Rodrigues, Douglas Torquato, João Paulo Madeiro, José Maria Monteiro, Angelo Brayner, Vânia Vidal, Narciso Arruda, Tiago Vinuto 41

Workshop on Thesis and Dissertations in Databases

Combining Meta-heuristics and Linear Programming to Address Ontology Meta-matching Problem
Nicolas Ferranti, Stênio Sã Rosário Furtado Soares, Jairo F. De Souza 47

Alinhamento de Grandes Ontologias com Recurso de Banco de Dados NoSQL e Utilização de Workflow Científico
Luciana de Sá Silva Perciliano, Fernanda Araujo Baião Amorim 53

Effective Method for Detecting Drunk Texting
Marcos A. Grzeça, Karin Becker, Renata Galante 60

Publicação de Dados Abertos Conectados Sobre os Transplantes Realizados no IMIP
Rayelle Ingrid Vera Cruz Silva Muniz, Bernadette Farias Lóscio 67

Versionamento de Conjuntos de Dados Publicados na Web
Wilker Cavalcante do Rego Santos, Bernadette Farias Lóscio 74

Caracterização e Comparação de Campanhas Promovendo o Outubro Rosa e o Novembro Azul no Twitter
Roberto Walter, Karin Becker 81

A Framework for Identification and Monitoring of Profiles and Behaviors of Users Based on Mobile App Usage
Nielsen Luiz Rechia Machado, Duncan Dubugras Alcoba Ruiz 88

An Autonomous Hybrid Data Partition for NewSQL DBs
Geomar André Schreiner, Denio Duarte, Ronaldo dos Santos Mello 95

Avaliação da Saúde de Ecossistemas de Dados
Glória de Fátima Andrade Barros Lima, Bernadette Farias Lóscio, Marcelo Iury S. Oliveira 102

A Process for Reverse Engineering of Aggregate-Oriented NoSQL Databases with Emphasis on Geographic Data
Angelo Augusto Frozza, Ronaldo dos Santos Mello 109

Partitioning Very Large de Bruijn Graphs for Genome Assembly
Julio O. Prieto Entenza, Sérgio Lifschitz 116

Author Index **123**

SBBD 2018

Demos and Applications

Lathe: Light-Weight Keyword Query Processing over Multiple Databases

Pericles de Oliveira[1], **Altigran da Silva**[1], **Edleno Moura**[1], **Gilberto Santos**[1]

[1] IComp/UFAM

{pericles.oliveira}@nokia.com, alti,edleno,gilberto@icomp.ufam.edu.br

Abstract. *We present a new R-KwS system called* Lathe*, which, is able to efficiently deal with such conditions. Lathe is based on a probabilistic model and principled strategies which allow pruning unlike candidate solutions for a keyword query very early in the process. As a result, the system is able to produce accurate answers very quickly in comparison to current systems. Lathe is able to handle many distinct databases by trying a same query over multiple different target databases at once, and providing answers for all those databases that have an answer to the query.*

1. Introduction

Among the most well-known existing R-KwS systems, there are those that take advantage of the basic functionality of the underlying RDBMS, and produce, from the keywords supplied in the queries, relational join expressions that try to fulfill the user's information needs. These relational expressions, called *Candidate Networks of Relations*, or simply *Candidate Networks* (*CNs*) [Hristidis et. al., 2002], are coded as SQL queries which, when executed in a target RDBMs, are expected to produce answers relevant to the keyword query posed by the user. Formally, CNs are joining trees of relation derived from a graph representing the schema of the target database. Thus, systems that rely on them are usually called *Schema Graph* R-KwS systems. Examples of these systems are DISCOVER [Hristidis et. al., 2002], CNRank [de Oliveira et al., 2015] and KwS-F [Baid et. al., 2010] [1].

We present a new Schema Graph R-KwS system called *Lathe*[2][3], which is able to efficiently deal with such conditions. Lathe is based on a probabilistic model and principled strategies, which allow pruning unlike Candidate Networks for a keyword query very early in the process. As a result, the system is able to produce accurate answers very quickly in comparison to current systems. Thus, our system represents a viable alternative to overcome a critical issue for the wide adoption of R-KwS systems in real applications

Lathe is able to seamlessly handle multiple target databases. Given a query, it is able to try a same input query over many databases at once, and providing answers for all those databases that are likely to have relevant results to the query. This not only simplifies the process of publishing new databases on-line, but also empowers the user to explore different databases from which she barely known the domain.

[1] There are in fact many more important systems that we did not cite only for saving space.

[2] This name alludes to the fact that the system automatically assigns an structure to an unstructured input keyword query, that is, it "shapes" the query.

[3] Funding by projects TTDSW (FAPEAM/CNPq), e-Vox Pesquisa (FAPEAM), e-Spot (CNPql), and by individual CNPq grants.

In the rest of this paper, we first present an overview on Lathe's architecture in Section 2. Finally, Section 3 describes how the demo session will be carried out.

2. System Overview

Lathes' architecture is illustrated in Figure 1. Given an input query, Lathe, as all other

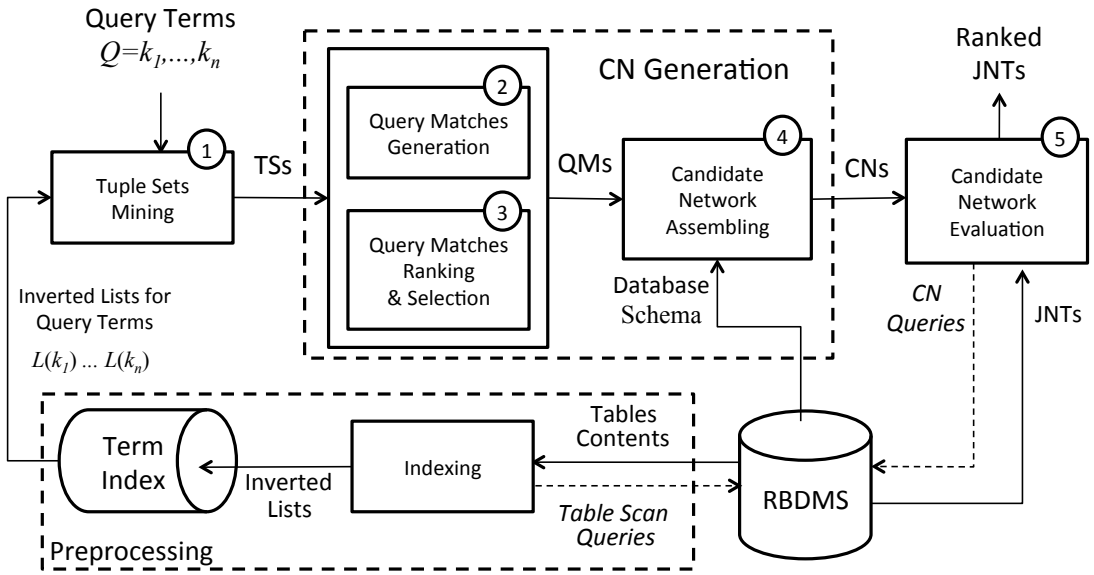


Figure 1. Lathe: architecture and main functioning.

Schema Graph R-KwS systems, first generates a number of Candidate Networks and then evaluates these CNs to produce answers. The main diference between Lathe and all previous systems lies in how it generates CNs. So far, all existing systems in the literature use for this task the CNGen algorithm proposed in [Hristidis et. al., 2002] for the DISCOVER system. In some cases, there is an additional step to somehow improve the set of CNs generated, as it occurs in KwS-F [Baid et. al., 2010] and CNRank [de Oliveira et al., 2015]. Lathe adopts a new approach for generating CNs, as we detail below.

Intuitively, a CN is a relational join expression that "connects" subsets of relations from the database whose tuples contain one or more keywords of the query. The "connections" are derived from referential integrity, i.e., PK/FK, constraints, which may involve additional relations. By having a DMBS to evaluate CNs, we obtain semantically meaningful answers as joined tuples which contain the query keywords.

As an example, consider the query "*denzel washington gangster*" and suppose we want to execute it over a relational database containing data on movies from the well-known Internet Movie Databases (IMDb). A possible CN for this query is given by:

$$\sigma_{\text{name}\supseteq\{gangster\}}\text{CHAR} \bowtie_{\text{id=cid}} \text{CAST} \bowtie_{\text{cid=pid}} \sigma_{\text{name}\supseteq\{denzel,washington\}}\text{PER} \quad (1)$$

where CHAR stores names of movie characters, PER stores data on persons (i.e., actors, actresses, directors, etc.) and CAST associates persons to the characters they play on movies. The join conditions in this expression are derived from PK/FK constraints.

Following the terminology introduced in [Hristidis et. al., 2002], the operands of the join operations in a CN are called *Tuple Sets*. Operands whose tuples contain the keywords specified in the query, such as those defined by selection operations over CHAR and PER in Equation 1, are called *Non-free Tuple Sets*. The remaining operands, such as CAST in Equation 1, are called *Free Tuple Sets*, since they not contain any of the keywords. For simplicity, in this paper we use *Tuple Sets* to refer to *Non-free Tuple Sets*, while *Free Tuple Sets* are referred explicitly.

The complexity of the CN generation problem is mainly due to two factors: (1) There can be multiple tuple sets for each subset of terms of the query. As a consequence, there may be a large number of ways of combining these tuple sets so that all terms of the query are covered; (2) Given a set of tuple sets that cover the terms of the query, there can be many distinct ways of connecting them through PK/FK constraints and free tuple sets.

3. Demonstration Plan

For the demo session, we prepared experiments using databases previously used in evaluations of R-KwS systems to highlight Lathe's main features. The experiments consist of tasks in which a user submits a keyword query, and, for this query, the system generates a number of Candidate Networks (CNs) and then evaluates these CNs to produce a ranking of Joint Networks of Tuples (JNTs) as a final answer. The goal is to demonstrate the behavior of the system in two aspects: the quality of the CNs produced, along with their impact on the quality of the results obtained when these CNs are evaluated; and the system's performance and scalability.

In each task, the user enters a keyword query and receives the ranked JNT provided by the system, with an indication of the time elapsed. The system also allows the user to examine details of each step of the keyword query processing, as described in Section 2 (see Figure 1). This includes the presentation of intermediary results, with summaries that demonstrate the behavior of the systems in terms of effectiveness and performance. The steps and the results that can be examined for each query execution are described in Table 1.

Table 1. Steps to be explored in the demo session.

Step	Results Presented
TS Mining	A list of the TSs are mining from termsets; a summary with the number of TSs generated and the time elapsed.
QM Generation	A list of QMs generated; a summary with the number QMs generated and the time elapsed.
QM Ranking	A ranking of the QMs generated; a summary with values for metrics such as precision, recall, MAP and RR, and the time elapsed.
CN Assembling	A list of the CNs generated from each QM; a summary with the number of CNs generated and the time elapsed.
CN Evaluation	The issuing of each CN to be evaluated on the DMBS; a summary with the number of JNTs obtained and the time elapsed.
JNT Ranking	The top-10 JNT; a summary with values for metrics such as precision, recall, MAP and RR, and the time elapsed.

We use 4 datasets: IMDb, Mondial, Wikipedia and DBLP, which were previously used for the experiments used in [Coffman et. al., 2010], [Luo et. al., 2007], [de Oliveira et al., 2015] [de Oliveira et al., 2018] and many other previous work. In Table 2 we present some details on these datasets, including their size (in Megabytes), the number of relations, the total number of tables and the number of Referential Integrity Constrains (RIC) in their schemas.

Table 2. Datasets used in the demo session.

Dataset	Size (MB)	Relations	Tuples	RIC
Mondial	9	28	17,115	104
IMDb	516	6	1,673,074	4
Wikipedia	550	6	206,318	5
DBLP	40	6	878,065	6

Queries used in the tasks were selected from 3 query sets from different sources: (1) Coffman: Queries used in the experiments reported in [Coffman et. al., 2010], targeted to datasets IMDb, Mondial and Wikipedia; (2) SPARK: Queries used in the experiments reported in [Luo et. al., 2007], targeted to datasets IMDb, DBLP and Mondial; (3) INEX: Queries originally specified for the INEX 2011 challenge[4], targeted to datasets IMDb which were adapted for searching in relational databases.

For didactic purposes, a companion poster showing the steps being executed is used to illustrate the running of the tasks. Attendees will also be invited to propose open, i.e., non-predefined tasks. Furthermore, besides trying the system during the session, interested attendees will also be able to experience the on-line system themselves on their on devices.

The demo session will be conducted on-line through a browser interface. The system is hosted on the cloud, in a regular machine with not parallel processing, running Centos 6.3 Linux. We use PostgreSQL as the underlying RDBMS with a default configuration. All implementations use the Java language.

Acknowledgments Work partially funded by projects eSpot (CNPq 461231/2014-0), SocSens (CAPES/PCGI 88887.130299/2017-01), CARECO (PROCAD/CAPES 88881.068507/2014-01), ATMOSPHERE (EC/H2020 grant no. 777154 & RNP/MCTIC acordo 51119), and by authors' individual grants from CNPq.

References

Baid et. al. (2010). Toward scalable keyword search over relational data. *PVLDB.*, 3(1-2):140–149.

Coffman et. al. (2010). A framework for evaluating database keyword search strategies. In *CIKM*, pages 729–738.

de Oliveira, P., da Silva, A. S., and de Moura, E. S. (2015). Ranking candidate networks of relations to improve keyword search over relational databases. In *ICDE*, pages 399–410.

de Oliveira, P., da Silva, A. S., de Moura, E. S., and Rodrigues, R. (2018). Match-based candidate network generation for keyword queries over relational databases. In *ICDE*, pages 16–19.

Hristidis et. al. (2002). DISCOVER: keyword search in relational databases. In *VLDB*, pages 670–681.

Luo et. al. (2007). SPARK: Top-k keyword query in relational databases. In *SIGMOD*, pages 115–126.

[4] `https://inex.mmci.uni-saarland.de`

VP-Viewer: Keeping track of your query from a vantage point*†

Daniel L. Jasbick[1], Thaylon Guedes[2], Rodolfo A. Oliveira[1], Lúcio F. D. Santos[3], Daniel de Oliveira[2], and Marcos V. N. Bedo[1]

[1]Fluminense Northwest Institute – Fluminense Federal University (UFF)

{danieljasbick, rodolfooliveira, marcosbedo}@id.uff.br

[2]Institute of Computing – Fluminense Federal University (UFF)

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

[3]Federal Institute of Technology North of Minas Gerais (IFNMG)

lucio.santos@ifnmg.edu.br

Abstract. *VP-Tree is the metric sibling of Binary Tree and k-Dimensional Tree indexing structures. However, visual exploration of VP-Trees is an open, yet important, issue as drawing crisp borders of VP-Tree partitions over a 2D transformation of metric data is often unachievable. In this demonstration, we present VP-Viewer, an index visualization tool for the investigation of the VP-Tree structure and the inspection of query paths. VP-Viewer builds upon a metric space library and enables the construction of parameterized VP-Trees, in which methods for distance calculation, pivot selection, and index balancing, besides the datasets themselves, are provided by the users. VP-Viewer renders VP-Trees by distinguishing directory nodes, which include vantage points, partition characteristics, and pivot-based distance distributions, from leaf nodes, which encompass the data elements and their distance to vantage points. Accordingly, users can easily explore the partitioning of a dataset for distinct parameterizations. Finally, VP-Viewer also enables the submission of range and kNN queries so that users can evaluate the tree branches examined by the searching algorithms.*

1. Introduction

Similarity searching is a base paradigm for the handling of data that are "alike" but not "equal". Such paradigm supports a variety of computational tasks, such as distance-based classification and content-based retrieval [Chávez et al. 2001, Padmanabhan and Deshpande 2015]. In practice, two of the most requested similarity searches are the range and neighborhood queries. An example of a range query in the bioinformatics domain is **(Q1)** Select all polypeptide chains that are different from a given chain by at most 3 codons, whereas a neighborhood (kNN) query example in the biomedical domain is **(Q2)** Find the 15 images of Magnetic Resonance Imaging (MRI) from distinct studies which are the most similar to a given MRI image of an (undiagnosed) patient. Range and neighborhood queries can be modeled upon *metric spaces*, where the

*The authors thank FAPEMIG, FAPERJ, CNPq and CAPES for their financial support.

†https://github.com/Jasbick/VP-Viewer

elements (polypeptide chains and MRI images, in the aforementioned examples) are represented as points and the (dis)similarity between each pair of points is evaluated by a distance function that complies to the symmetry, positivity and triangle inequality properties [Hetland 2009, Padmanabhan and Deshpande 2015].

Several metric access methods have been proposed to speed up similarity-based queries [Chávez et al. 2001, Chen et al. 2017]. Such methods accelerate similarity searches by targeting an optimization criterion, such as the number of disk accesses, the number of distance calculations, or the overhead caused by the searching algorithm [Chávez et al. 2001, Hetland 2009]. *Vantage-Point Trees* (VP-Trees) access methods are particularly versatile for enhancing query executions, as they enable the organization of the search space in a hierarchical and disjoint fashion [Li et al. 2014]. VP-Trees are indexing structures that extend the concept of a Binary Trees for the querying of metric spaces within logarithm time complexity once each decision of searching either a left or right node may halve the number of subtrees to be evaluated [Yianilos 1993].

Roughly speaking, VP-Trees organize data from a set $\mathcal{S}$ using a *pivot* p, a median μ of distances from p to elements in $\mathcal{S}$, a maximum distance d_m between p and any element in $\mathcal{S}$, and two partitions generated from p: *left* and *right* nodes. Elements whose distances to p fall inside the $[0, \mu)$ interval are assigned to the left node, whereas elements of the $[\mu, d_m]$ interval are set to the right node. The left and right nodes are datasets themselves and can be recursively divided until either each node becomes a unitary set, or a maximum number of elements per leaf node is reached. The last criterion generates the VP-Tree variation called vp^{sb}-tree, which we shall examine hereafter. Median μ is a careful choice for the disjointed partitioning of $\mathcal{S}$ as unique medians would split the dataset into a perfectly balanced tree. Such *uniqueness*, however, depends on the distance distribution so that the resulting tree may be unbalanced if leaf nodes are unable to handle overflow. Likewise, the method for selecting VP-Tree pivots directly affects tree balancing and branching-based search quality [Li et al. 2014]. VP-Tree pivot set includes the data elements that *maximize* the variance of the distance distribution [Hetland 2009], but the solution for fetching such an optimal set is polynomial [Ruiz et al. 2013]. Alternatively, several heuristics can be used for reducing the pivot selection costs for large databases, such as *Randomness*, *Sampling*, and *Convex Hull Points* [Chávez et al. 2001].

Finding the most suitable setting of a VP-Tree, *i.e.*, pivot selection and tree balancing, is often unintuitive and experimentally burdensome [Li et al. 2014, Chen et al. 2017]. In this demonstration, we present VP-Viewer, a tool for assisting users in both understanding and assessment of VP-Trees. Although existing applications, *e.g.*, the C++-based MAM-View [Chino et al. 2010] or a JavaScript-based web solution[1], can be used for the visualization of indexed metric data, they focus on rendering 2D representations of data partitioning and can express neither the relationship between VP-Tree nodes nor the distance distributions within rooting nodes. Unlike these previous approaches, VP-Viewer distinguishes VP-Tree *directory* nodes, which include vantage points, partition lower and upper bounds, covered number of elements, and pivot-based distance distributions, from *leaf* nodes, which encompass the data elements and their distance to vantage points. Accordingly, users can easily explore the partitioning of a dataset for distinct parameterizations. Furthermore, VP-Viewer is not only limited to generate VP-Tree visualizations.

[1] `https://fribbels.github.io/vptree/vptree.html`

Our tool also enables the user to submit range and neighborhood queries so that they can interactively evaluate the tree nodes that were examined by the VP-Tree branch-and-bound algorithms, *i.e.*, users can compare distinct VP-Tree searching performances.

2. The VP-Viewer Architecture

VP-Viewer is a cross-platform desktop application that assists users in the exploration and assessment of VP-Trees. Internally, VP-Viewer is composed of a set of connected modules, as in Figure 1. The tool builds upon the Arboretum metric space library[2] for *(i)* reading user-provided datasets, and *(ii)* casting them into a common abstraction regardless of the domain (number, string, etc.) – Figure 1(1–3). VP-Viewer provides its own implementation of VP-Trees by employing both Arboretum data abstraction and metric distance function interface. We also employ open-source Graphviz library[3] for the graph representation of the resulting VP-Tree structure – Figure 1(4).

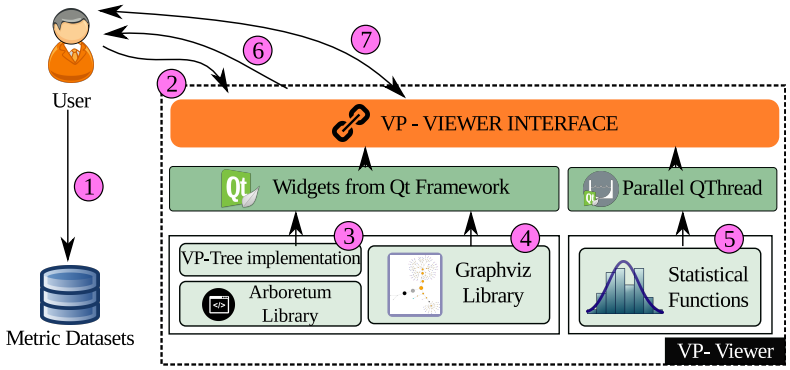


Figure 1. Overview of VP-Viewer's architecture.

Qt framework[4] is used for the incorporation of both VP-Tree implementation and Graphviz graph representation into *widgets*, which are ready-to-use GUI components. We also implemented a group of routines for the selection of pivots and for the gathering of pivot-based distance distributions – Figure 1(5). Such routines are implemented by extending the Qt thread interface so that they run along with the VP-Tree construction. Finally, we integrated all resources into a single GUI interface that supports data loading, zoom-in/out, inspection of pivot-based distance distributions within directory nodes, and submission-and-solving of range and neighborhood queries – Figure 1(6 – 7). The visual exploration of similarity searches involves the interaction between VP-Tree implementation, Graphviz library, and Statistical Functions modules.

In the first step, VP-Tree implementation executes a branching-based algorithm for the query execution and labels the evaluated tree nodes, whereas the Statistical module gathers the number of both distance calculations and inspected nodes. Next, while Graphviz renders the labeled nodes by highlighting them, the differences between the branching-based costs are juxtaposed to the brute-force solution (sequential scan) costs in as a bar percentage chart. Finally, the highlighted query path and the searching bar costs are embedded into widgets and displayed to the user.

[2]https://www.bitbucket.org/gbdi/arboretum
[3]https://www.graphviz.org/
[4]https://www.qt.io/

3. Demonstration of VP-Viewer

Here, we use VP-Viewer on three real-world datasets aiming at providing an exemplification of the scenarios covered by our tool. Table 1 details the demonstration datasets regarding cardinality, dimensionality and distance function (δ).

Table 1. Description of the datasets used in the demonstration.

Dataset	Card.	Dim.	δ	Description	Available at
CITIES	$5,507$	2	L_2	Lat-long coordinates of $5,507$ Brazilian cities.	www.ibge.gov.br
WORDS	$5,757$	–	$Edit$	All data elements are English words composed of precisely 5 characters.	www-cs-faculty.stanford.edu/~knuth
ISOLET	$6,238$	617	L_1	Features extracted from records of persons speaking each alphabet letter.	archive.ics.uci.edu/ml/datasets/isolet

Upon accessing VP-Viewer, users can visualize the main panel (Figure 2) that requests the disk location of the dataset to be indexed (1), the data cardinality (2) and dimensionality (3), the maximum allowed number of elements per leaf node (4), the distance function to be used (5), the pivot selection method (6), and, finally, the authorization parameter for node overflow (7) that may be set to "True" if balanced tree is a hard constraint, or "False" otherwise. At this point, users can request the VP-Tree construction by clicking on "generate" button and explore the resulting structure. Although VP-Viewer includes L_1, L_2, and $Edit$ functions, other distances can be plugged in through the Arboretum library interface. VP-Viewer alternatives for pivot selection are "Random", "Yianilos's Sampling", and "Convex Hull".

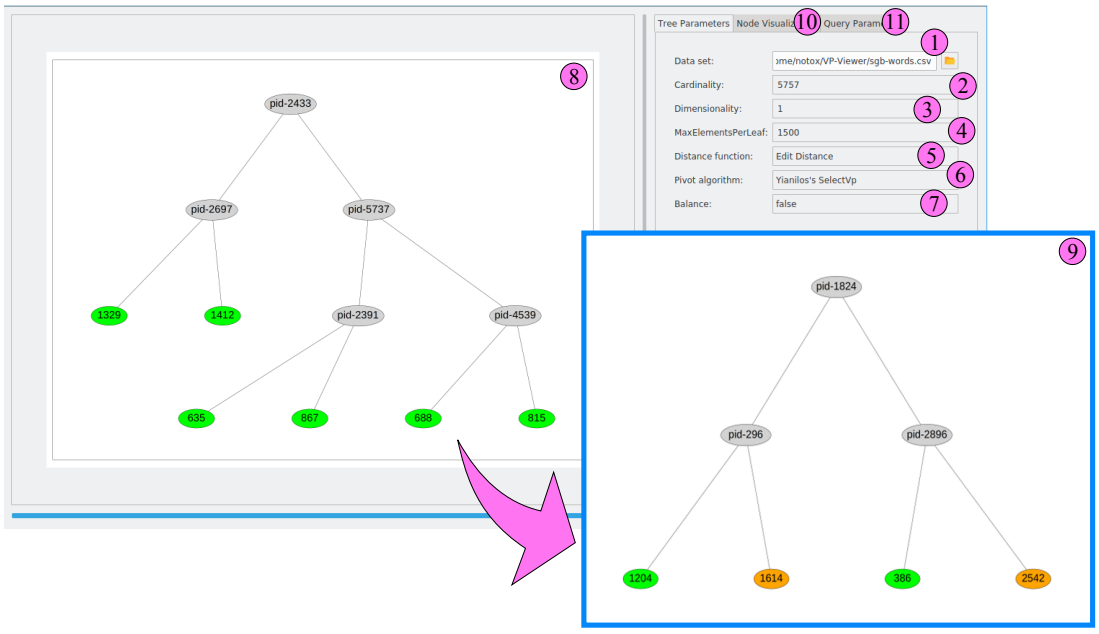


Figure 2. VP-Viewer main interface.

Figure 2 illustrates two different VP-Trees constructed for WORDS dataset with distinct constraints on node overflow. In this scenario, tied elements at the median distance to the pivots are expected because the $Edit$ distance returns discrete measures in the $[0, 5]$ interval. Figure 2(8) shows the resulting VP-Tree for a relaxed overflow constraint, which means VP-trees are allowed to be unbalanced. Green nodes indicate all leaves contain no more than the user-specified number of elements. Alternatively, a balanced tree requires the right children of directory nodes to handle overflow, as in Figure 2(9). Orange nodes indicate *(i)* leaves are currently storing more elements than specified in the user parameters, and *(ii)* additional disk paging may be necessary whenever the number of elements is related to the disk page size.

Before changing the VP-Tree parameterization for softening the tie issue, users can verify the *pivot-based distance distribution* on the VP-Viewer Node Visualization tab – Figure 2(10)). Such a resource enables the user to explore the distance distribution within each directory node. By selecting a particular node, users can visualize the distance histogram of the elements rooted by the directory node to its pivot. Figure 3(a) shows an example for the unbalanced case of WORDS dataset. Orange histogram bar is the bucket of the median and expresses the probability of ties in terms of frequencies. The probability is recursively propagated to the right-most node of the structure.

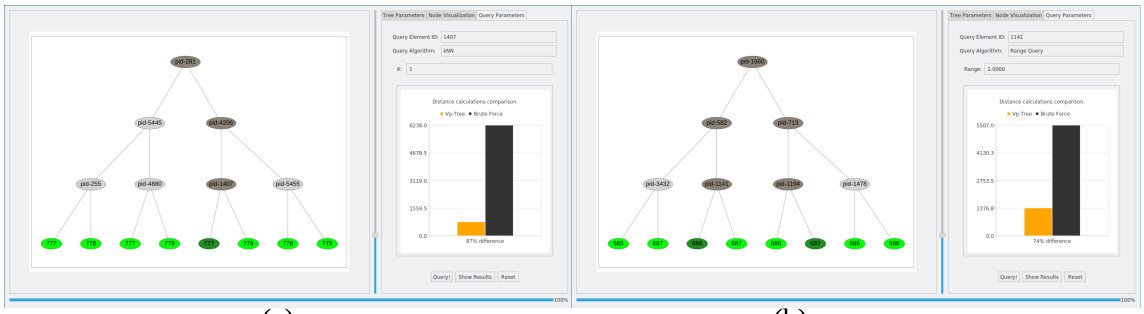
Although, choosing a new pivot selection method may diminish (or derail) the ties of median values, distance concentration around medians can be unavoidable for certain cases. For instance, Figure 3(b) shows the distance distribution for a VP-Tree directory node constructed for the ISOLET dataset with Yianilo's sampled pivots, 800 elements per leaf node, and unbalanced tree constraint setting. In this high-dimensional case, the majority of pivot-based distance distributions resemble the Standard distribution in which median behaves analogously to the mean. The switch of Yianilo's sampled pivots to either Random or Convex Hull criteria did not change the Standard distribution behavior of distances in the evaluations we performed.

(a) (b)

Figure 3. Exploration of pivot-based distance distribution. (a) WORDS dataset. (b) ISOLET dataset.

The last aspect we consider for VP-Viewer is experimental evaluation may be necessary for finding the most suitable VP-Tree partitioning. VP-Viewer enables users to request range and neighborhood queries on the Query parameters tab – Figure 2(11)). Figure 4(a) provides an example of a neighborhood query on ISOLET dataset, whereas Figure 4(b) shows an example of a range query on CITIES dataset. VP-Viewer executes both branching-based VP-Tree search and brute-force algorithms for each requested

query, and presents the performance differences between the two searching routines as a bar plot. Our tool also labels the inspected nodes that are highlighted as the *query path* in the main interface. As a result, users can visualize and interpret VP-Tree searching parameters and also compare distinct VP-Trees settings by evaluating their performances over a (sequence of) similarity query.



(a) (b)

Figure 4. Similarity searching on VP-Viewer. (a) A neighborhood query on `CITIES` dataset. (b) A range query on `ISOLET` dataset.

4. Conclusions

In this demonstration, we presented VP-Viewer, a tool that supports users in the understanding and assessment of VP-Trees. VP-Viewer is composed of a set of individual modules, which enable users the coupling and testing of several methods for distance calculation and pivot selection. We examine the capabilities of VP-Viewer and discuss usage scenarios for three real-world datasets. Our application has presented interesting resources in the handling of metric data from different perspectives.

References

Chávez, E., Navarro, G., Baeza-Yates, R., and Marroquín, J. L. (2001). Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321.

Chen, L., Gao, Y., Zheng, B., Jensen, C. S., Yang, H., and Yang, K. (2017). Pivot-based metric indexing. *PVLDB*, 10(10):1058–1069.

Chino, F. J. T., Vieira, M. R., Traina, A. J. M., and Jr., C. T. (2010). MAMView: A Framework for Visualization of Metric Trees. In *SBBD – Demo Section*, pages 1–6.

Hetland, M. L. (2009). The basic principles of metric indexing. In *Swarm Intelligence for Multi-objective Problems in Data Mining*, pages 199–232. Springer.

Li, Q., Z., H., Lei, F., L., G., Lu, M., and Mao, R. (2014). Excluded Middle Forest vs. VP-Tree: An Analytical and Empirical Comparison. In *PAIS*, pages 431–437. Springer.

Padmanabhan, D. and Deshpande, P. M. (2015). *Operators for Similarity Search - Semantics, Techniques and Usage Scenarios*. Springer.

Ruiz, G., Santoyo, F., Chávez, E., Figueroa, K., and Tellez, E. S. (2013). Extreme pivots for faster metric indexes. In *SISAP*, pages 115–126. Springer.

Yianilos, P. N. (1993). Data structures and algorithms for nearest neighbor search in general metric spaces. In *ACM-SIAM SDA*, pages 311–321. SIAM.

Metamorfose: a data transformation framework based on Apache Spark

Evandro Miguel Kuszera[1,2], Leticia M. Peres[2], Marcos Didonet Del Fabro[2]

[1]Federal Technological University of Paraná (UTFPR) – Dois Vizinhos – PR – Brazil

[2]Federal University of Paraná (UFPR) – C3SL Labs – Curitiba, PR – Brazil

{evandrokuszera@utfpr.edu.br, lmperes,marcos.ddf}@inf.ufpr.br

Abstract. *On the context of large availability of open data, it is important to provide interactive solutions where a data transformation workflow can be easily deployed and developed. This article presents Metamorfose[1], a framework for data transformation based on large-scale data processing engine Apache Spark. Through the presented framework, the user can set up an interactive transformation workflow for loading data, defining mappings between source and target fields, performing transformations, and persisting the data. The user can define transformation functions in Java or Javascript and create chains of transformation where a transformation result can be used as input to the next.*

1. Introduction

The ability to exchange and integrate data is crucial for many types of applications, especially with the diversity of data formats (e.g., structured or semi-structured). With the emergence of open data, new solutions must be developed to easily explore and analyze the large amount of data generated. Data transformation systems (DTS) are important in these scenarios. The function of these systems is to perform transformations where instances of a source schema are translated to instances of a target schema. DTSs are composed of a set of translation tasks that can be defined as a quadruple $\{S, T, I_{\mathrm{s}}, I_{\mathrm{e}}\}$, where S is the source schema, T is the target schema, I_{s} is a valid instance of S and I_{e} a valid instance of T generated by applying the desired transformations over I_{s} [Mecca et al. 2012]. A set of mappings M is used to denote the relationship between expected output and provided input.

Schema mapping and ETL (*Extract*, *Transform*, *Load*) tools are widely used to transform and integrate diverse sources of data. Transformations in schema mapping tools are based on declarative specifications (such as SQL, for example), not providing the exact method they are implemented and executed. ETL tools define transformations as data flows over a graph of operators. Operators range from simple mapping between tables to complex data join and splitting operations, where the user can choose a specific implementation to a data transformation [Dessloch et al. 2008].

CLIO [Haas et al. 2005] was one of the pioneering tools to execute schema mappings. Others have recently been proposed as ++Spicy [Marnette et al. 2011], LLunatic [Geerts et al. 2014] and EXLEngine [Atzeni et al. 2017]. Clover ETL [CloverETL 2018], Pentaho Kettle [Pentaho Kettle 2018], OpenRefine [OpenRefine 2018] and Talend Open

[1]Metamorfose video: https://youtu.be/ta9mXuCeIwM

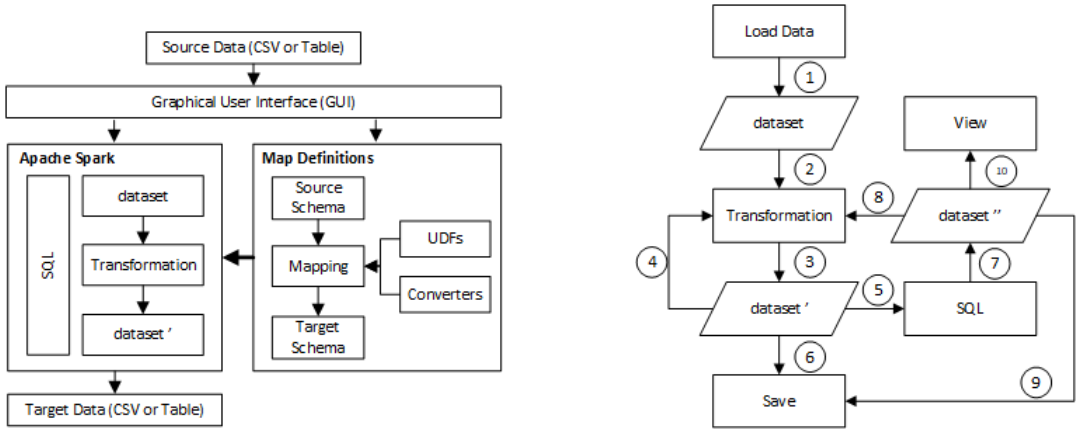
Studio [Talend 2018] are examples of ETL tools. Schema mapping tools are intended for designers who need to express the main components of a data transformation task. While ETL tools are intended for users interested in efficient implementations of a data transformation task. Depending on the context, the use of these tools is not trivial, requiring a complex installation and configuration process to perform simple data transformations. In these scenarios concise tools are an option, not requiring the mastery of a large set of technologies to perform data transformations.

In this paper we present Metamorfose, a framework for data transformation based on large-scale data processing engine Apache Spark [Zaharia et al. 2016]. The choice of using Apache Spark was motivated by its support of a wide range of processing workloads and abstractions provided to manipulate datasets. Through Metamorfose the user can load data, define mappings between source and target fields, execute transformations, and persist data. Metamorfose is extensible and independent of the underlying data format. New data sources can be added by creating Spark datasets. In the current version, it can load and persist data as CSV file or relational database table. As a contribution, the framework allows the definition of an interactive transformation workflow that can be integrated with user-defined transformation functions implemented in Java or Javascript. The use of Javascript makes it flexible to add new transformation functions. In the next section the framework will be presented.

2. Metamorfose

The architecture of Metamorfose and mapping definitions will be presented below.

2.1. Architecture of Metamorfose



(a) Architecture of Metamorfose. (b) Execution Flow.

Figure 1. Architecture and Execution Flow of Metamorfose.

The Figure 1 (a) presents the main components of Metamorfose. Through the graphical user interface it is possible to load data source for transformation, execute data queries through Spark SQL (Apache Spark module) and specify mappings between source and target schema. Each data source is loaded as a Spark dataset. To transform the data, map functions are executed on the source dataset according to the user-defined mappings (Map Definitions module).

The Metamorphosis execution flow over one data source is presented in Figure 1 (b). In (1) the data is loaded as a Spark dataset. Based on the user mappings the dataset is transformed (2) producing (3) dataset' as a result. Dataset' can be used: (4) as input to a new transformation, (5) as input to SQL query or (6) can be persisted to CSV file or relational database table. The SQL query execution on dataset' produces new (7) dataset". Dataset" can be used for new data transformations (8), for persistence (9) or visualization (10). This flow allows to flexible define chains of transformations. Transformations can be applied over the current dataset or can create a new dataset with the resulting data.

2.2. Mapping Definitions

A data schema can be defined as $S = \{s_1, ..., s_n\}$, where s_i represents a data field of the schema S. From a source schema S is possible to derive a target schema T applying transformations on the field values of S. Transformations can be represented as a set of mappings $M = \{(s_1, t_1, f_1), ..., (s_n, t_n, f_n)\}$, where s_i represents one or more source fields, t_i a target field and f_i a transformation function. Transformation chains can be defined using the target schema T as the source schema for next transformation. To persist T data for an existing data source, T must correspond to the data source schema.

Figure 2 (a) shows an example of transformation over a table with person data. ID field is mapped to the COD field in the target schema. NAME and LASTNAME fields are mapped to the NAME field. SEX field data are transformed from 'F' and 'M' to numerical values 1 and 2, respectively. ADDRESS and PHONE fields only exist in the target schema and they will receive an empty string and null value, respectively.

ID	NAME	LASTNAME	SEX
1	João	Silva	M
2	Maria	Silva	F
3	José	Silva	M

COD	NAME	SEX	ADDRESS	PHONE
1	João Silva	1	''	null
2	Maria Silva	2	''	null
3	José Silva	1	''	null

(a) Source and target schemas.

#	SOURCE	TARGET	TYPE	SCRIPT
1	ID	COD	Casting	
2	$LIST(NAME, LASTNAME)	NAME	Function	function concat (value) { return value[0] + ' ' +value[1]; }
3	SEX	SEX	Function	function sex (value) { if (value[0]=='F') return 1; else if(value[0]=='M') return 2; }
4	$VALUE('')	ADDRESS	Casting	
5	$VALUE(NULL)	PHONE	Casting	

(b) Transformations settings.

Figure 2. Mapping and transformation example.

Mappings are declarative statements that establish correspondence between source and target fields, which can be integrated with data transformation functions. Figure 2 (b) shows the mappings used to transform the data. $VALUE() and $LIST() are reserved words and act as Metamorfose operators. Line 1 displays a one-to-one mapping between source and target, with *Casting* data type conversion. In line 2 the word $LIST denotes a many-to-one mapping, where a list of source fields is sent to user-defined function (UDF) *concat*. In this simple example a concatenation operation is performed. However, complex functions can be defined in Javascript. In line 3 is an example of one-to-one mapping using UDF *sex*. In lines 4 and 5 the word $VALUE is used to insert constant values into the target schema. This set of mappings can be persisted as a JSON file for future use.

3. Demonstration

The Metamorfose tool provides a graphical user interface to ease the definition of mappings and data transformation execution. In the demonstration we will present the main flow of tool: data loading, mapping definitions and transforming execution.

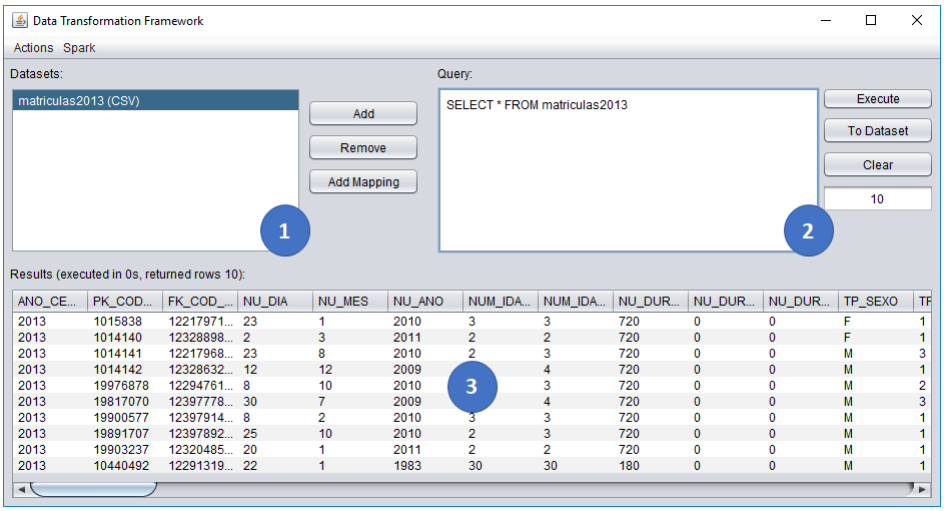


Figure 3. Metamorfose main screen.

3.1. Data Loading and Exploration

The Figure 3 presents the Metamorfose main screen. In (1) the user can load or remove datasets, view the list of available datasets, or add mappings to selected dataset. *Add Mapping* button shows the mapping screen (details in the next section). The user can load datasets from CSV files or relational database tables (Postgres, for example). In (2) the user can submit SQL queries (using Spark SQL) on the loaded datasets. In (3) the user can view and explore the data produced by the SQL query. SQL query results can be added as a new dataset in the list of available datasets pressing *To Dataset* button in (2). In the example of Figure 3 a CSV file (*matriculas2013*) with four million records was loaded and a SQL query was submitted to visualize the data in the result table.

3.2. Mappings

The user must select a dataset from the list and click the *Add Mapping* button to define mappings. The mapping screen appears and the selected dataset scheme (source) is loaded automatically. The Figure 4 presents a mapping previously defined and loaded from a JSON file. In (1) the fields and data types of the source dataset are displayed. In (2) the user must specify the fields and data types of the target dataset. In (3) the user must define the type of the transformation (*Casting* or *Javascript*). For transformations involving Javascript UDFs the user must enter the code in the edit screen in (4). After mapping definitions between source and target fields, two options are available (5): the *Apply Mapping* button applies the mappings over the source dataset and modify your data and schema, or the *Apply Mapping to New Dataset* button creates a new dataset from source dataset. The latter option allows it to define transformation chains. The new datasets are added in the list of available datasets for further use. In Figure 4 there are mappings for field renaming, insertion of constant values in the target dataset and *Javascript* UDFs where one or more source fields are transformed to one target field according with UDF logic.

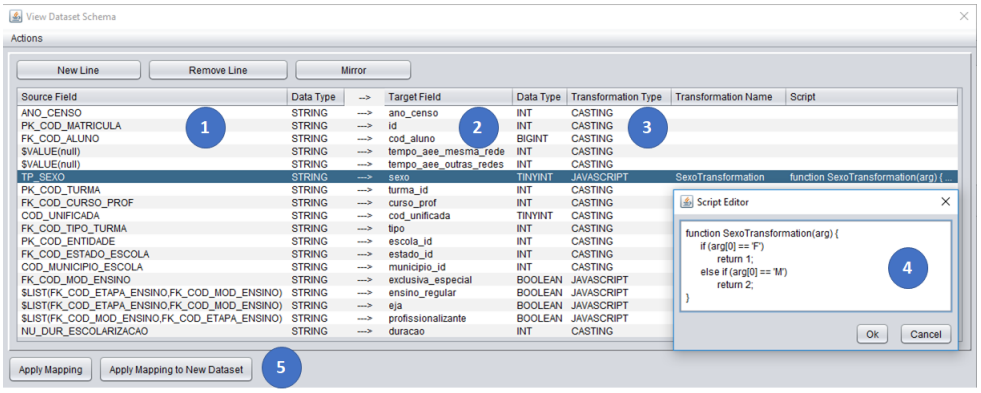


Figure 4. Metamorfose mapping definition screen.

3.3. Data Transformation

After defining the mappings the user can visualize or persist the contents of the dataset through Metamorfose main screen (Figure 3). The persistence options are CSV and JDBC and are available through the *Actions* menu. At this point the data transformations are executed by Apache Spark.

4. Evaluation

Yearly, the National Institute of Educational Studies and Research Anísio Teixeira (INEP)[2] provides open data about educational census in Brazil. These data including enrollments, teachers, schools and classes in CSV format that exceed 69 million records per year. This data source can be used to analyze the situation of education in Brazil. However, the format of data is not rigid, variations are found in every release, such as adding new fields, changing and removing existing fields, for example. Metamorfose was used to map and transform CSV data to a predefined PostgreSQL database.

We have performed experiments using Brazilian enrollment data from the year 2013. Due to the number of records, the data were made available in five CSV files representing the South, Southeast, Midwest, North and Northeast regions of the country (around 55 million records). It were defined 82 mapping fields between CSV file and target relational schema. One-to-one, many-to-one mappings and Javascript UDFs were used to transform data.

Table 1. Number of records and execution time of experiment performed.

Brazil Region	Records	Time
Midwest	4.038.979	10 min
South	7.276.108	18 min
Northeast	16.729.543	41 min
Southeast + North	27.379.690	65 min

Metamorfose was run on a machine with Intel Core i7 2.5GHz processor, 16GB RAM, Windows 10 Home and Postgres 10. The Table 1 shows the number of records by region of Brazil and execution time to transform and load the data to relational database.

[2]INEP: http://www.inep.gov.br/

This experiment aimed to validate the tool on a data set of moderate size. It is possible to observe that the execution time has linear growth in respect of the number of records. Future experiments will be carried out considering more records and processing nodes.

5. Conclusions

In this paper we have presented the Metamorfose tool, a framework for data transformation built on large-scale data processing engine called Apache Spark. Through a graphical user interface is possible to define an interactive data transformation workflow where the user can loading data, defining mappings, performing transformations, exploring and persisting the results. A mapping definition can be integrated with data transformation functions implemented in Java or Javascript, providing a flexible way to define complex transformations. It is possible to execute SQL queries to filter, aggregate and join the datasets before or after every data transformation. The results can be persisted as CSV file or as relational database table. Metamorfose was validated using real open data about education in Brazil. As future work, new features will be added including transformation functions, supporting for other data sources (JSON and XML) and support for other database models.

References

Atzeni, P., Bellomarini, L., Bugiotti, F., and De Leonardis, M. (2017). Executable Schema Mappings for Statistical Data Processing. *Distributed and Parallel Databases*.

CloverETL (2018). http://www.cloveretl.com/. Accessed on 6 January 2018.

Dessloch, S., Hernandez, M. A., Wisnesky, R., Radwan, A., and Zhou, J. (2008). Orchid: Integrating schema mapping and etl. In *2008 IEEE 24th ICDE*, pages 1307–1316.

Geerts, F., Mecca, G., Papotti, P., and Santoro, D. (2014). That's All Folks! Llunatic Goes Open Source. *PVLDB*, 7(13).

Haas, L. M., Hernández, M. A., Ho, H., Popa, L., and Roth, M. (2005). Clio grows up: From research prototype to industrial tool. In *Proceedings of the 2005 ACM SIGMOD*, SIGMOD '05, pages 805–810, New York, NY, USA. ACM.

Marnette, B., Mecca, G., Papotti, P., Raunich, S., and Santoro, D. (2011). ++spicy: an opensource tool for second-generation schema mapping and data exchange. *PVLDB*, 4(12).

Mecca, G., Papotti, P., Raunich, S., and Santoro, D. (2012). What is the iq of your data transformation system? In *Proc. of the 21st ACM CIKM*, CIKM 2012, pages 872–881.

OpenRefine (2018). http://openrefine.org/. Accessed on 4 January 2018.

Pentaho Kettle (2018). https://www.hitachivantara.com/. Accessed on 4 April 2018.

Talend (2018). https://www.talend.com/. Accessed on 5 Febuary 2018.

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., and Stoica, I. (2016). Apache spark: A unified engine for big data processing. *Commun. ACM*, 59(11):56–65.

Análise *Online* de Dados de Proveniência e de Domínio de Aplicações Spark com SAMbA*

Thaylon Guedes[1], Vítor Silva[2], Marcos V. N. Bedo[1], Marta Mattoso[2], e Daniel de Oliveira[1]

[1]Universidade Federal Fluminense (UFF)

{thaylongs, marcosbedo}@id.uff.br, danielcmo@ic.uff.br

[2]Universidade Federal do Rio de Janeiro (COPPE/UFRJ)

{silva,marta}@cos.ufrj.br

Resumo. *Usuários dos mais diversos domínios de aplicação são capazes de se beneficiar de frameworks para processamento paralelo, escalável e intensivo em dados como o Apache Spark. Entretanto, o Apache Spark não oferece apoio à captura e gerência de dados de proveniência que auxiliam na depuração, reprodutibilidade e análise dos resultados. Um dos desafios em adicionar dados de proveniência ao Spark é capturar dados de proveniência distribuídos em estruturas em memória (RDDs) e associá-los ao histórico de derivação de dados. Esta demonstração apresenta o SAMbA, uma extensão para o Spark capaz de coletar dados de proveniência dos RDDs em tempo de execução e fornecer recursos analíticos ao usuário. Na demonstração utilizamos como estudo de caso uma execução de consultas por similaridade de forma paralela.*

1. Introdução

Muitas aplicações (científicas e comerciais) são compostas por operações que produzem dados de forma intensiva, como cargas, transformações e agregações de dados [Liu et al. 2016]. O fluxo dos dados gerados pelo encadeamento dessas operações é geralmente representado por meio de grafos direcionados acíclicos ou DAGs (do inglês *Directed Acyclic Graph*), no qual os vértices representam transformações de dados enquanto os arcos representam as dependências de dados entre elas. Várias aplicações consomem e produzem um grande volume de dados e adotam *scripts* para implementar tais DAGs. Entretanto, uma vez que essas aplicações necessitam processar conjuntos de dados de grande cardinalidade, o processamento paralelo, distribuído e escalável se torna essencial para o desempenho da aplicação [Jha et al. 2014].

Uma maneira simplificada de transformar *scripts* sequenciais em paralelos é por meio do uso de *frameworks* DISC (do inglês *Data Intensive Scalable Computing*). O Apache Spark[1] é um DISC que provê execução paralela e escalável de *scripts* com uso intensivo de dados. O Spark é baseado no processamento distribuído de dados em memória, por meio de uma abstração chamada RDD (do inglês *Resilient Distributed Dataset*). Os RDDs são coleções distribuídas de elementos de dados imutáveis. Uma vez modelada a aplicação, o Spark analisa o fluxo de dados da mesma para coordenar a execução paralela de instruções nas partições de dados que compõem um RDD.

*Os autores gostariam de agradecer a CAPES, CNPq e FAPERJ por financiarem parcialmente o trabalho

[1]https://spark.apache.org

Um exemplo de aplicação que requer processar grandes volume de dados é a execução de consultas por similaridade em espaços métricos. Esse tipo de consulta envolve, por exemplo, uma classificação baseada em distância ou recuperação de dados por conteúdo. Na prática, os dois tipos de consultas por similaridade mais utilizadas são as buscas por abrangência e vizinhança [Hetland 2009]. A modelagem e representação de operadores de buscas por similaridade em espaços métricos vem sendo estudada por muitos anos [Padmanabhan and Deshpande 2015] e diferentes estratégias de indexação têm sido propostas para otimizar o desempenho de sua execução, tais como os índices VP-Trees e VP-Forests. Estes índices são evoluções de árvores binárias e visam a executar buscas por similaridade com, potencialmente, complexidade logarítmica de tempo [Yianilos 1998]. Entretanto, a depender da cardinalidade e dimensionalidade do conjunto de dados de entrada, a quantidade de nós-folhas a ser examinada pode ser não-negligenciável, o que torna a execução da busca custosa. Portanto, consultas por similaridade (mesmo indexadas) podem se beneficiar do processamento paralelo do Spark.

Apesar de escalável para o processamento de grandes quantidades de dados, o Spark carece de apoio à proveniência [Freire et al. 2008]. Dados de proveniência são fundamentais para registrar a linhagem da geração de dados e facilitar a reprodutibilidade e a análise de dados produzidos por uma aplicação. Quando os dados de proveniência são enriquecidos com os dados do domínio da aplicação, eles se tornam uma base de dados importante para a análise *online* e *post-mortem*. Em execuções de longa duração, as consultas *online* são essenciais para monitorar e analisar o andamento da aplicação. O Spark fornece uma capacidade de proveniência limitada que registra as atividades já executadas em forma de arquivos log, mas é incipiente no que tange a prover apoio à reprodutibilidade e à análise de dados. Além disso, a depuração do código do Spark também é bastante difícil, dada a sua característica paralela e distribuída. Sistemas como `BigDebug` [Gulzar et al. 2016] e `Titian` [Interlandi et al. 2015] surgiram para fornecer recursos de depuração ao Spark, mas se restringem ao registro de dados de execução sem se preocuparem com a análise dos dados.

Este artigo tem por objetivo apresentar o SAMbA[2] (Spark provenAnce MAnagement), uma extensão para tornar o Spark capaz de coletar dados de proveniência a partir dos RDDs e fornecer recursos analíticos em tempo de execução. Samba supera os diversos desafios com relação à captura e registro de dados de proveniência no Spark, sendo um deles a de capturar dados de proveniência distribuídos nos RDDs e associá-los em memória ao caminho de derivação de dados de proveniência. Outro desafio relevante ocorre quando as transformações de dados no Spark trocam dados por meio de leitura e gravação de dados em arquivos. Nesse contexto, a transferência de elementos de dados entre atividades é implícita e o Spark não pode usar os RDDs para gerenciar esses dados porque não está ciente do fluxo de dados. Portanto, capturar os dados desses arquivos tem imenso potencial de melhorar a análise de dados de proveniência, mas também é um desafio [Freire et al. 2008, Silva et al. 2017].

2. Arquitetura do SAMbA

A captura e a gerência dos dados de proveniência de aplicações Spark não é uma tarefa simples. Primeiramente, os usuários devem ser capazes de definir o conteúdo apropriado

[2]`https://github.com/UFFeScience/SAMbA`

(elementos de dados de interesse) a serem extraídos dos RDDs. Além disso, aplicações Spark são geralmente executadas em ambientes DISC (nuvens ou *clusters*) e os usuários não têm ciência de qual máquina executará cada atividade. O SAMbA aborda ambos os desafios, coletando proveniência prospectiva e retrospectiva [Freire et al. 2008], além de informações do ambiente de execução e dos dados específicos do domínio. A proveniência prospectiva representa as operações que são executadas sobre os dados na aplicação, ao passo que a proveniência retrospectiva representa o *log* de execução da aplicação. O registro das informações do ambiente descrevem características relacionadas ao ambiente DISC. Por fim, os dados específicos de domínio representam o conteúdo produzido pelas transformações (resultados intermediários) que precisam ser analisados em conjunto com os dados de proveniência.

A Figura 1 apresenta os principais componentes da arquitetura do SAMbA. Os componentes do Spark coloridos são estendidos por meio do SAMbA e funcionam conforme discutido na sequência: o componente *Spark Context* conecta-se ao Gerenciador de *clusters* (Figura 2). Assim, o Spark instancia *Executors*, que são processos que executam operações e armazenam dados nas máquinas do ambiente DISC. Depois o Spark envia o código da aplicação (arquivo executável) para os *Executors*. Em seguida, o *SparkContext* envia tarefas para os *Executors* para serem processadas. Desse modo, o SAMbA é executado em cada *Executor* do Spark para capturar os dados de proveniência e de domínio.

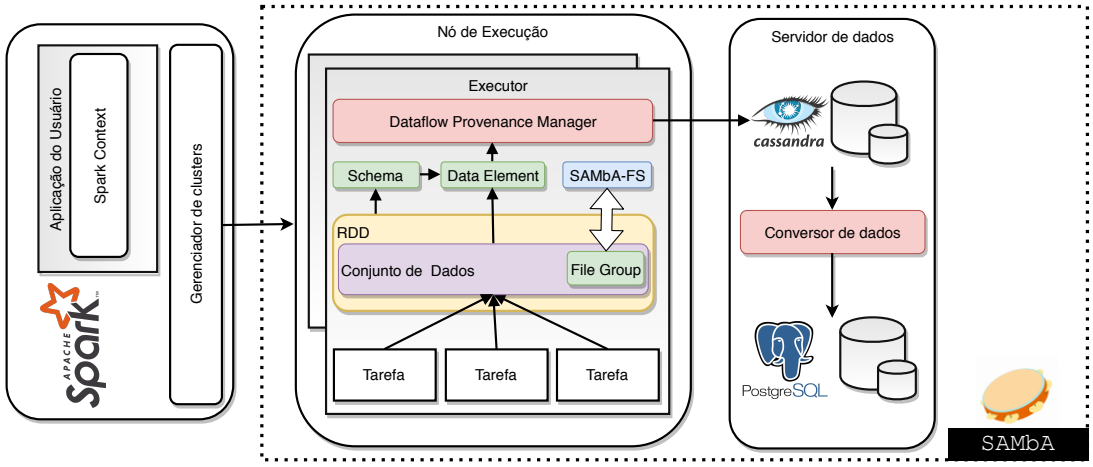


Figura 1. Arquitetura do SAMbA.

```
/* Spark Context */
val sparkConf = new SparkConf().setAppName("RangeQueryVpTree").setMaster("local[4]")
val sc = new SparkContext(sparkConf)
val forest: Broadcast[util.List[Node[Array[Double]]]] = sc.broadcast(rangeArgs.getArgsParser.getTrees)
```

Figura 2. Definição do *Spark Context* no SAMbA

O fluxo de um nó em execução na Figura 1, gerenciado pelo *Dataflow Provenance Manager*, registra a proveniência prospectiva e captura a proveniência retrospectiva do *dataflow* gerada pela execução da aplicação Spark. Essa captura de dados é baseada na extração de dados do RDD, através do *Data Element*, o qual representa cada unidade de dados que compõem o conteúdo de um RDD. Para adicionar dados de domínio à proveniência do *dataflow*, o SAMbA permite que o usuário defina atributos de interesse para

serem extraídos usando um componente chamado *Schema*. Assim, o *Dataflow Provenance Manager* interage com o *Data Element*, aplicando sobre ele o *Schema* (Figura 3).

```
.setSchema(new DataElementSchema[(RangeQueryParameter, String, Double)] {
  override def getFieldsNames(): Array[String] = Array("Target Element Id", "Candidate Element ID", "Distance")
```

Figura 3. Exemplo de uso do *Schema* no SAMbA.

O SAMbA concentra-se no usuário que desenvolveu a aplicação Spark, e que também conhece os dados do domínio. Assim, o *Schema* é uma *interface* com os métodos `getFieldsNames()`, que define os atributos a serem representados no *Schema*, e `splitData()`, que retorna uma coleção de dados (*Data Collection*) extraída do RDD. Um *Data Collection* é um conjunto de *Data Elements*, onde cada *Data Element* possui um identificador único (ID). Este ID é usado pelo SAMbA para rastrear a proveniência dentro do banco de dados que a armazena. Da mesma forma, cada operação executada no SAMbA também possui um ID, que é usado para associar o *Data Element* com as operações que o geraram.

Para resolver o problema de capturar dados de proveniência de programas caixa-preta, o SAMbA fornece o SAMbA-FS, um sistema de arquivos baseado em libfuse[3] (*Filesystem in Userspace*) do Linux, que mapeia um certo diretório do usuário para os dados representados por um RDD em memória. Como resultado, o Spark, e consequentemente o SAMbA, ficam cientes dos arquivo manipulados por programas caixa-preta. O SAMbA-FS utiliza o tipo de dado *FileGroup* que representa um conjunto de arquivos em memória. Para criar um RDD de *FileGroup*, utilizamos uma classe utilitária chamada *FileGroupTemplate*, e através dela, definimos quais arquivos devem ser carregados em memória – como exemplificado na Figura 4. Neste caso, o conteúdo do arquivo chamado *inputFastaList.txt* é carregado na memória (RDD). Além disso, o *FileGroupTemplate* permite armazenar um mapa de chave-valor, que serve para armazenar informações extras sobre o *FileGroup*, como o nome do arquivo ou algum parâmetro de configuração.

```
val fileGroupTemplate = FileGroupTemplate.ofFile(
      new File("/home/user/dataflow/inputFastaList.txt"),
      false, Map("FILE_NAME" -> "inputFastaList.txt")
)
val rdd = sparkContex.fileGroup(fileGroupTemplate)
```

Figura 4. Criação de um RDD de *FileGroup*.

Quando um RDD de *FileGroup* é criado, dois novos operadores de transformação são fornecidos pelo SAMbA: *runCommand*, que executa comandos nativos do SO, e *runScientificApplication* que executa programas ou *scripts* caixa-preta contendo sequência de invocações de programas (Figura 5) que se encontram em um determinado diretório.

```
rdd.runScientificApplication("someScript.sh {{FILE_NAME}}")
```

Figura 5. Exemplo da execução de um *script* como um parâmetro *FILE_NAME* proveniente do mapa de chave-valor do FileGroup.

[3] `https://github.com/libfuse/libfuse`

Somando-se a essas características, o SAMbA carrega dados de proveniência e domínio em seu banco de dados de forma *online*, ou seja, durante a execução de aplicações Spark. Desta forma, a proveniência e os dados de domínio são fornecidos para o usuário em tempo de execução, permitindo assim que os usuários executem consultas enquanto a aplicação ainda está em execução. Os dados de proveniência capturados são armazenados em um banco de dados Apache Cassandra pelo *Dataflow Provenance Manager*.

3. Demonstração

Em nossa demonstração, exemplificamos como o SAMbA pode ajudar os usuários a executar aplicações Spark e realizar análises *online* sobre os dados específicos de domínio armazenados em memória. Para tanto, utilizamos como estudo de caso uma aplicação que paraleliza consultas por similaridade indexadas em espaços métricos. A ideia principal desse estudo de caso é acelerar a avaliação dos elementos nos nós-folha de índices VP-Tree e VP-Forest ao resolver uma consulta por similaridade.

Nesse contexto, os nós-folha incluem os elementos de dados a serem consultados e são armazenados como arquivos sequenciais, enquanto que as estruturas das árvores em si são mantidas como arquivos separados. Dada uma consulta por abrangência com um raio de tolerância e um elemento de busca, definimos uma rotina Spark para selecionar os nós-folha que interceptam o raio da consulta e examinar a distância das instâncias cobertas pelo nó com relação ao elemento de consulta. Nesse cenário, exemplos de consultas *online* que podem ser executadas pelos usuário para acompanhar a execução são: (1) Verificar quais foram os nós-folhas que foram inspecionados, ou (2) Verificar quantos nós foram descartados por não incluírem elementos candidatos ao conjunto-resposta.

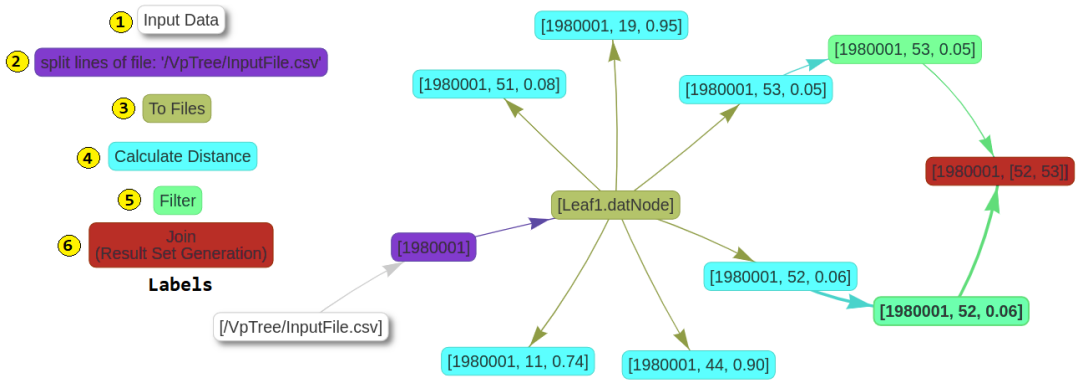


Figura 6. Grafo de proveniência do SAMbA para busca por abrangência em um nó-folha de um índice VP-Tree.

Além de executar essa aplicação no Spark, os usuários podem visualizar o grafo de proveniência gerado (Figura 6). Para este estudo de caso, usamos conjuntos de dados sintéticos de 10 e 100 dimensões comparados pela distância Euclideana e usando índices com diversas parametrizações no número de elementos por nó-folha. O grafo apresentado na Figura 6 representa a execução do dataflow para uma consulta por abrangência cujo o raio seja menor ou igual a 0,06 unidades. A aplicação do estudo de caso começa lendo o arquivo de entrada (1). Para cada elemento de consulta presente no arquivo de testes (2), são identificados os nós-folhas dos índices que talvez contenham elementos que atendam o critério de seleção. Cada elemento do nó-folha (3) é identificado pelo seu ID e tem a

sua distância calculada até o elemento de consulta. Em seguida, são selecionados aqueles cuja a distância é menor ou igual a 0,06 unidades (4). Por fim, a aplicação agrupa os identificadores dos elementos que passaram no teste (5). Para a apresentação do SAMbA no evento, incentivamos usuários a trazerem suas próprias aplicações Spark.

4. Conclusão

O SAMbA é uma extensão do Apache Spark para apoiar a análise *online* de dados de proveniência armazenados em memória. Nesta demonstração apresentamos um exemplo do uso do SAMbA para captura e consulta de proveniência *online* em uma aplicação de busca por similaridade. Como trabalhos futuros, pretende-se investigar o uso do SAMbA no gerenciamento de proveniência em aplicações científicas com relação a consultas *online* e *post-mortem*. A versão beta do SAMbA pode ser obtida no repositório `https://github.com/UFFeScience/SAMbA`. Além disso, está disponível no mesmo endereço do repositório, um vídeo de demonstração do sistema e um guia inicial para usuários que desejam utilizar o SAMbA.

Referências

Freire, J., Koop, D., Santos, E., and Silva, C. T. (2008). Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3):11–21.

Gulzar, M. A., Interlandi, M., Yoo, S., Tetali, S. D., Condie, T., Millstein, T., and Kim, M. (2016). Bigdebug: Debugging primitives for interactive big data processing in spark. Em *Proceedings of the 38th International Conference on Software Engineering*, páginas 784–795. ACM.

Hetland, M. L. (2009). The basic principles of metric indexing. Em *Swarm Intelligence for Multi-objective Problems in Data Mining*, páginas 199–232. Springer.

Interlandi, M., Shah, K., Tetali, S. D., Gulzar, M. A., Yoo, S., Kim, M., Millstein, T., and Condie, T. (2015). Titian: Data provenance support in spark. *Proceedings of the VLDB Endowment*, 9(3):216–227.

Jha, S., Qiu, J., Luckow, A., Mantha, P., and Fox, G. C. (2014). A tale of two data-intensive paradigms: Applications, abstractions, and architectures. Em *IEEE International Congress on Big Data*, páginas 645–652.

Liu, J., Pacitti, E., Valduriez, P., de Oliveira, D., and Mattoso, M. (2016). Multi-objective scheduling of scientific workflows in multisite clouds. *Future Generation Computer Systems*, 63:76 – 95. Modeling and Management for Big Data Analytics and Visualization.

Padmanabhan, D. and Deshpande, P. M. (2015). *Operators for Similarity Search - Semantics, Techniques and Usage Scenarios*. Springer.

Silva, V., Leite, J., Camata, J. J., de Oliveira, D., Coutinho, A. L., Valduriez, P., and Mattoso, M. (2017). Raw data queries during data-intensive parallel workflow execution. *Future Generation Computer Systems*, 75:402 – 422.

Yianilos, P. N. (1998). Excluded middle vantage point forests for nearest neighbor search. Technical report, NEC Research Institute, Princeton, NJ.

ImgDW Generator[1]: A tool for generating data for medical image data warehouses

Guilherme Muzzi da Rocha[1], Cristina Dutra de Aguiar Ciferri[1]

[1]Departamento de Ciências de Computação – Universidade de São Paulo
Caixa Postal 668 – 13.560-970 – São Carlos – SP – Brasil

guilherme.muzzi.rocha@usp.br, cdac@icmc.usp.br

Abstract. *In this paper, we introduce ImgDW Generator, a tool that generates synthetic data for populating medical image data warehouses designed according to the relational technology. The tool supports different star schemas for the image data warehouse and offers a graphical interface that assists users to manipulate these schemas. ImgDW Generator can be used to generate data aiming at different scenarios of performance evaluation.*

1. Introduction

Data warehousing environments (DWEs) play a key role in decision-making [Chaudhuri et al. 2011]. They support an ETL (extract-transform-load) process aimed to extract, transform, clean, integrate, and load data from heterogeneous sources into large databases called data warehouses (DWs). In addition to these characteristics, data in the DWs are subject-oriented, non-volatile, and referring to large periods of time. In relational implementations of DWs, data are usually modeled through a star schema, where a central fact table is linked to several satellite dimension tables. DWEs also support OLAP (on-line analytical processing) queries, which are complex analytical queries aimed to discover useful trends and patterns. Conventional DWEs only manage conventional data, such as alphanumeric and numeric types.

Image DWEs extend conventional DWEs to deal with images. Because there is no agreement in the literature about the definition of these environments, we use the principles introduced by Teixeira et al. (2015). Instead of managing images as matrices of pixels or files in the DICOM (Digital Imaging and Communications in Medicine) format, images are represented through their intrinsic features, i.e. *feature vectors* and *attributes for similarity search*. As a result, image DWEs should support an extended ETL process that also performs the extraction of features from images, an extended DW (i.e. an *image DW*) that also stores images features in fact or dimension tables, and an extended query processing that enables OLAP similarity queries.

Image DWEs empower decision-making by enabling users to issue a new range of queries, which integrate the conventional OLAP and the similarity search processes. For instance, consider an application related to the medical field that stores images of exams in addition to conventional data related to patients and years. Using these environments, specialists are able to issue queries such as “How many images are similar to a given cancer image, considering patients with ages between 40 and 50, and years from 1992 to 2018?”. Assessing the performance of this new range of queries is

[1] https://www.lsec.icmc.usp.br/medicalimage. The authors thank the financial support of the following Brazilian agencies: CAPES, CNPq, FAPESP and FINEP.

not an easy task. It is very difficult to generate image data and to relate them with conventional data.

This paper introduces ImgDW Generator, a tool for generating data for medical image DWs implemented according to the relational technology. We have designed ImgDW Generator to provide the major characteristics as follows.

- It offers a graphical and interactive interface.
- It supports four different star schemas to model the medical image DW.
- It generates conventional and image data related to the medical field to populate the medical image DW.

To the best of our knowledge, there is no other tool that generates intrinsic features of images and relates them with conventional DW data. This paper is organized as follows. Section 2 describes background, Section 3 details ImgDW Generator, Section 4 shows configurations for performance evaluation, and Section 5 concludes the paper.

2. Background

The computational management of images requires the construction of an image descriptor, which is described by two aspects [Traina et al. 2007]. The first one is an extraction algorithm that encodes image features into feature vectors. Common image features are attributes of color and texture. Well-known image extractors for these attributes are Color Histograms [Gonzalez and Woods 2006] and Haralick descriptors [Haralick 1979]. The feature vectors generated by these extractors contain the numeric representations of the images, and are usually represented in the metric space. Thus, the second aspect refers to a distance function, which calculates the dissimilarity between two images based on their feature vectors. A distance function becomes smaller as the images become more similar, thus enabling the execution of similarity search.

Environments that manage large volumes of image data should improve similarity search performance by pruning portions of the database where a given query image cannot be found. To this end, the principles introduced by Teixeira et al. (2015) use the Omni-technique to select strategically positioned images from the dataset, called representative images [Traina et al. 2007]. The number of representative images depends on the intrinsic dimensionality of the dataset according to the applied image extractor. For instance, Color Histograms may require three representative images.

Figure 1 depicts four different star schemas that may be modeled to store medical images in an image DW, according to the results described by Annibal (2011). The fact table (e.g. Exam) contains foreign keys to the dimension tables and numeric measures; conventional dimension tables (e.g., Age) store a primary key and several descriptive conventional data; and image dimension tables (e.g., Histogram) store a primary key and distances to representative images. The schemas differ on how they manage the storage of the feature vectors, as described as follows: (i) *attributes of a joint dimension table* (Figure 1a): feature vectors are stored as attributes in a unique dimension table, which may contain feature vectors related to distinct image dimension tables; (ii) *attributes of a single dimension table* (Figure 1b): feature vectors are stored as attributes in a single dimension table, which cannot contain feature vectors related to distinct image dimension tables; (iii) *facts in the fact table* (Figure 1c): feature vectors are stored

as numeric measures in the fact table, which may contain feature vectors related to distinct image dimension tables; and (iv) *attributes of an image dimension table* (Figure 1d): features vectors are stored together with their respective image dimension table.

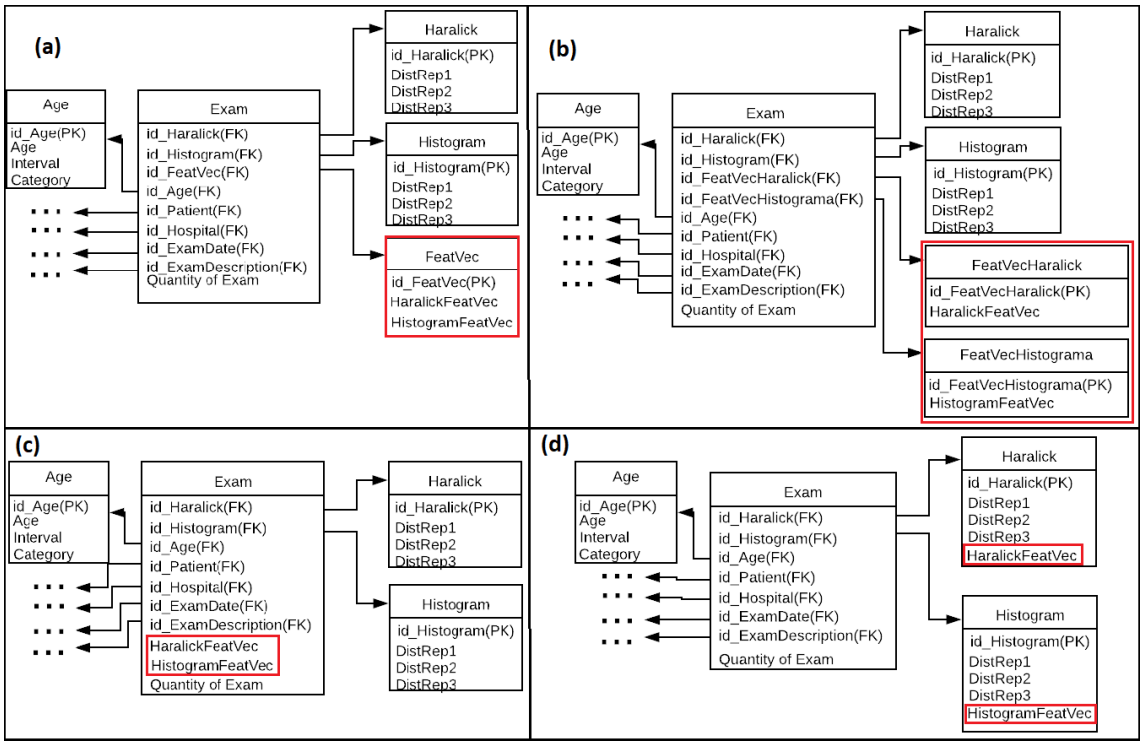


Figure 1. Four different star schemas to model a medical image DW [Annibal 2011]: (a) *attributes of a joint dimension table*; (b) a*ttributes of a single dimension table*; (c) *facts in the fact table*; and (d) *attributes of an image dimension table.*

3. ImgDW Generator

In this section, we introduce ImgDW Generator, a tool for generating data for medical image DWs implemented according to the star schemas depicted in Figure 1. The tool considers as **default application** a medical application that models the fact *Quantity of Exams*, considering the conventional dimensions *Age*, *Patient*, *Hospital*, *ExamDate*, and *ExamDescription*, and the image dimension tables *Histogram*, *Haralick1*, *Haralick2*, *Haralick3*, and *Haralick4*. Different Haralick descriptors allow analyzing different texture features of images [Haralick 1979].

Using the initial interface of ImgDW Generator, the user can set **configurations parameters**, such as the directory where data will be saved, and choose the star schema to be used. Because the tool works similarly for each star schema, here we consider that the user has chosen the schema *attributes of a joint dimension table* (Figure 1a) to show the major characteristics of ImgDW Generator.

Based on the user's option, ImgDW Generator displays the interface shown in Figure 2. The fact table is visually drawn in the center, conventional dimension tables are drawn in green on the left, and image dimension tables are drawn in blue on the right. The dimension table that stores the feature vectors is represented in red on the right. The user may iterate with the tool by choosing one of the following tags: generate data (Section 3.1) and generate create/insert script (Section 3.2).

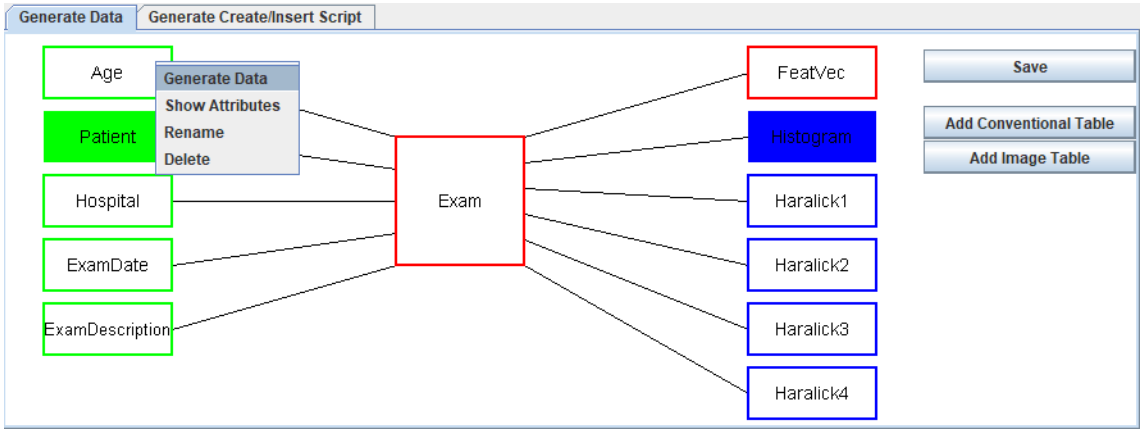


Figure 2. Interface of the tag Generate Data, considering the schema *attributes of a joint dimension table.*

3.1. Tag Generate Data

Tag Generate Data is used to generate synthetic data to populate the fact table, the conventional dimension tables, and the image dimension tables.

When the user chooses a given table, she can generate data, show its attributes, rename the table, and delete the table (Figure 2). Figure 3a depicts data generation for the conventional dimension table Age. The user can set the number of tuples to be generated by defining a maximum value or an interval of values, select the attributes to be generated, and add new attributes as needed. For each new attribute, the user should name it and select an external text file that contains data values for its domain. Figure 3b shows data generated for the table Age, which are stored in a CSV file. Similarly, Figure 4 shows an example of data generation for the image dimension table Histogram.

Data generation should follow an order: *first*, populate the conventional and image dimension tables; *second*, populate the feature vectors table; *third*, populate the fact table. ImgDW Generator visually illustrates that a table has been populated as follows. Before being populated, only the border of the table is colored (see the dimension tables Age and Haralick1 in Figure 2). After being populated, the table is fully colored (see the dimension tables Patient and Histogram in Figure 2).

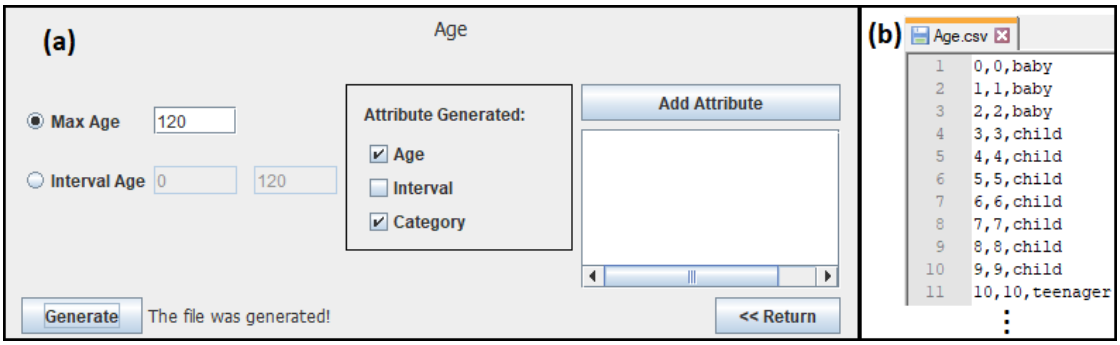


Figure 3. Example of data generation for the conventional dimension table Age: (a) interface with options; (b) generated CSV file.

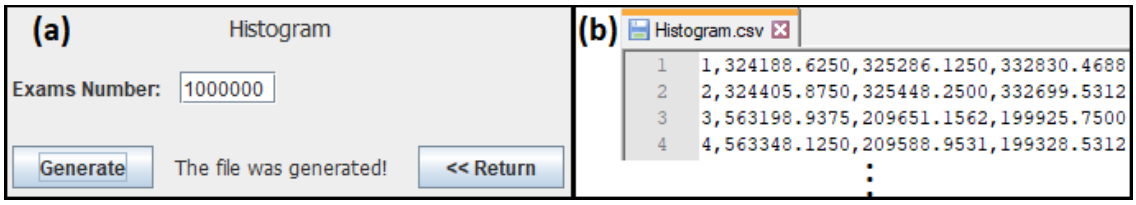


Figure 4. Example of data generation for the image dimension table Histogram: (a) interface with options; (b) generated CSV file.

At any time, the user can add new conventional and image dimension tables (Figure 2). To this end, she should define the type of the table, its name and its attributes. For each attribute, she also should select an external text file that contains the domain of values of the attribute. That is, the user can use specific data sets to populate tables. Another available functionality is saving the work.

For the default application, ImgDW Generator already encompasses external text files that contain data to populate each table. Although the tool generates synthetic data, it provides real data when possible. Data that populate the conventional dimension table Hospital are real and were obtained from the Brazilian health care system available at www2.datasus.gov.br/datasus/index.php. Data that populate the conventional dimension table ExamDate are also real, and refer to days from 1992 to 2018. Furthermore, image data were generated using feature vectors obtained from 1,000,000 real images from a Brazilian public hospital. The remaining dimension tables are populated with synthetic data mainly due to privacy issues. Regarding the fact table, it associates each image to its conventional data. It is a factless fact table since it contains as numeric measure the artificial attribute *quantity of exams*, which is always populated with the value of 1.

3.2 Tag Generate Create/Insert Script

Tag Generate Create/Insert Script is used to generate SQL commands to create the dimension and fact tables and to insert data into these tables. This tag is only available to the user when all tables of the star schema are populated.

The definition of the CREATE TABLE commands follows the characteristics of the star schema defined in the tag Generate Data, respecting the tables and attributes selected by the user. The insertion of data is performed using the COPY command and the generated CSV files. Figure 5 depicts the create/insert scripts for the conventional dimension table Age and the image dimension table Histograms. The directory and the delimiter specified in the COPY command are set in the configuration parameters. In its current version, ImgDW Generator is set to generate SQL scripts for PostgreSQL®.

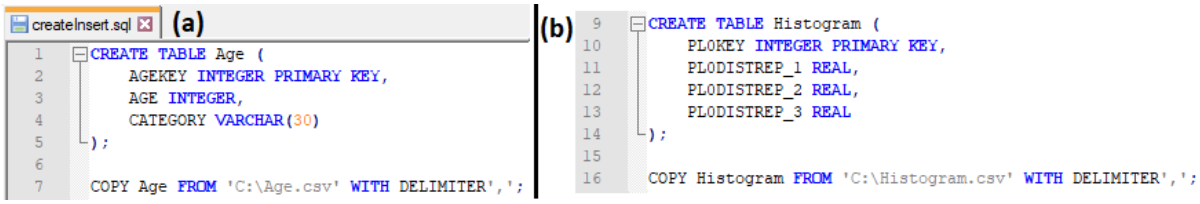


Figure 5. Create/insert scripts: (a) conventional dimension table Age; (b) image dimension table Histogram.

Table 1. Number of tuples generated by ImgDW Generator for scenarios 1 and 2.

Scenario 1				Scenario 2	
Tables	# tuples	Tables	# tuples	Tables	# tuples
Age	121	Histogram	1,000,000	Age	121
Patient	100,000	Haralick1	1,000,000	ExamDate	4,934
Hospital	645	Haralick2	1,000,000	Exam	500,000
ExamDate	9,868	Haralick3	1,000,000	Histogram	500,000
ExamDescription	1,000,000	Haralick4	1,000,000	Haralick1	500,000
Exam	1,000,000	FeatVec	1,000,000	FeatVec	500,000

4. Configurations for Performance Evaluation

ImgDW Generator can be used to generate data aiming at different scenarios of performance evaluation. For instance, consider the following two scenarios: (i) *scenario 1*, which is composed of all tables shown in Figure 2; and (ii) *scenario 2*, which is composed of only some of these tables. Table 1 depicts the tables and their respective number of instances (i.e. # tuples) for each scenario. Performance evaluation tests should use these medical image data warehouses to issue SOLAP similarity queries over them considering different dimensionalities and data volumes. Performance evaluation considering these scenarios can be found in Annibal (2011) and Teixeira et al. (2015).

5. Conclusions

In this paper, we introduce ImgDW Generator, a tool for generating data for medical image data warehouses implemented according to the relational technology. The tool offers a graphical and interactive interface through which users can work with different star schemas that model the medical image data warehouse; set conventional and image dimension tables, attributes and values for these attributes; and generate SQL scripts containing commands for creating and populating the data warehouse. ImgDW Generator was implemented in Java using NetBeans version 8.2, thus it is portable. We are currently extending the tool to support different database management systems. We are also developing new functionalities to assist users to define SQL commands that perform OLAP similarity queries over medical image data warehouses.

References

Annibal, L.P. (2011) Istar: Um Esquema Estrela Otimizado para Image Data Warehouses Baseado em Similaridade. Dissertação de Mestrado, UFSCar.

Chaudhuri, S., Dayal, U., Narasayya, V.R. (2011). An Overview of Business Intelligence Technology. Communications of the ACM, v. 54, n. 8, p. 88-98.

Gonzalez, R.C., Woods, R.E. (2006). Digital Image Processing. Prentice-Hall, 3rd ed.

Haralick, R.M. (1979). Statistical and Structural Approaches to Texture. Proceedings of the IEEE, v. 67, n. 5, p. 786-804.

Teixeira, J.W., Annibal, L.P., Felipe, J.C.; Ciferri, R.R.; Ciferri, C.D.A. (2015) A Similarity-based Data Warehousing Environment for Medical Images. Computers in Biology and Medicine, v. 66, p. 190-208.

Traina Jr., C., Santos Filho, R.F., Traina, A.J.M., Vieira, M.R., 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. v. 16, n. 4, p. 483-505.

Outer-Tuning: sintonia fina automática baseada em ontologia

Ana Carolina Almeida[1], Edward Hermann Haeusler[2], Sérgio Lifschitz[2], Rafael Pereira de Oliveira[2], Daniel Schwabe[2]

[1] Depto de Informática e Ciência da Computação UERJ
ana.almeida@ime.uerj.br

[2]Departamento de Informática PUC-Rio
{hermann,sergio,rpoliveira,dschwabe}@inf.puc-rio.br

Resumo. *Apresentamos neste trabalho o framework Outer-Tuning, que visa dar apoio à sintonia fina (semi) automática de sistemas de bancos de dados relacionais por meio de uma ontologia específica ao domínio. Neste trabalho apresentamos os principais aspectos de sua arquitetura baseada em componentes, a máquina de regras de inferência e uma visão geral da ferramenta na prática.*

1. Introdução

O trabalho de administração e sintonia fina (*tuning*) de bancos de dados é complexo e exige especialização e conhecimentos fundamentais nesta área da computação. Busca-se um melhor desempenho para o banco de dados, isto é, maior eficiência ou vazão (*throughput*) de suas transações. Para isso, realizam-se ajustes de suas configurações, parâmetros e projeto físico, seleção de estruturas de acesso, redundância de estruturas físicas, sempre de acordo com a carga de trabalho executada no banco.

Nosso grupo de pesquisa já desenvolveu a ferramenta *DBX*[1] para a manutenção automática e contínua do projeto físico de um banco de dados relacional. Neste artigo, apresentamos a arquitetura de *software*, aspectos funcionais e práticos do *framework* Outer-Tuning [Almeida 2013], uma ferramenta baseada em ontologia criada para apoiar DBAs e desenvolvedores em geral na tomada de decisão envolvida na atividade de sintonia fina.

O restante deste artigo está organizado conforme descrito a seguir. Na Seção 2 motivamos a construção da ferramenta e o uso de ontologias no processo de sintonia fina. A Seção 3 apresenta o projeto de arquitetura e a Seção 4 ilustra o funcionamento básico do *framework* aqui proposto. A Seção 5 conclui este trabalho.

2. Motivação: ontologia e alternativas para sintonia fina

O trabalho de [Almeida 2013] propôs originalmente o *framework* Outer-Tuning, visando apoiar o trabalho de sintonia fina (semi)automática em sistemas de bancos de dados relacionais (SBDR). O nome da ferramenta originou-se do fato da mesma apresentar todas as alternativas analisadas pelas heurísticas e não somente aquela considerada a melhor ação de sintonia fina. Faz-se uma analogia à operação relacional de *outer join*, que retorna todas as tuplas das tabelas envolvidas e não somente aquelas que satisfazem à condição de junção. A ideia básica do Outer-Tuning consiste em oferecer mecanismos que permitam ao DBA uma tomada de decisão mais consistente com relação às possíveis ações de sintonia fina, baseada em múltiplas alternativas e respectivos custos.

[1]https://github.com/BioBD/dbx

Nas ferramentas de sintonia-fina existe uma falta de clareza sobre as decisões e as ações que são tomadas de forma automática. Dessa forma, o Outer-Tuning propõe o uso de uma ontologia de aplicação para a sintonia fina (automática ou não) que proporciona uma abordagem formal para decisões e inferências. A contribuição inovadora dessa abordagem é oferecer transparência e confiabilidade acerca das alternativas disponíveis para possíveis cenários no SGBD, por meio de justificativas concretas para as decisões que foram tomadas definidas semanticamente.

Foram encontrados apenas dois trabalhos preocupados com semântica [Goasdoué et al. 2011][Khouri et al. 2012], mas ambos somente relacionados aos dados e não aos conceitos de sintonia-fina. Além disso, as ferramentas não são flexíveis o suficiente para incorporar novas heurísticas. Através do uso de uma ontologia específica busca-se gerar automaticamente novas práticas de sintonia fina, a partir das práticas existentes (uso de inferências) ou de novas regras e conceitos que venham a surgir no futuro. Esta abordagem permite também combinações de heurísticas de sintonia fina.

A partir da ontologia do domínio de sintonia fina, proposta em [Almeida 2013] e estendida em [Oliveira 2015], buscou-se atender os desafios de especificar os componentes e integrá-los com uma máquina de regras. Usaremos Visões Materializadas (VM) para ilustrar as vantagens do uso da ontologia (Figura 1) em todo o processo. Supondo um usuário *User_1* que submete um comando DML *DML_1*. Este é classificado automaticamente como sendo *DMLCommand - SingleStatement - QueryStatement* através de regras definidas na própria ontologia. Esse comando possui como propriedade *hasDescription* o comando SQL derivado do benchmark TPC-H[2]: *SELECT no_o_id FROM new_order WHERE no_w_id = 1 AND no_d_id = 1;*. Continuando na figura, a ontologia já infere, por regras, as cláusulas *SELECT*, *FROM* e *WHERE* definidas em tal comando. A ontologia de domínio completa pode ser encontrada na URL[3].

Um exemplo de regra definida na ontologia pode ser visto na Figura 2. Para utilizar uma determinada heurística de VM é necessário estimar o custo de criação da visão materializada. Este é obtido, através de regra SWRL definido na ontologia, como sendo o custo de uma varredura simples da visão materializada mais o custo de gravação das páginas em memória secundária, estimada como sendo equivalente a 2 (duas) vezes a quantidade de páginas hipotéticas da VMH [Oliveira 2015]. Dessa forma, tem-se que dado o conceito de VM hipotética (VMH), se existe alguma VMH (linha 1 - Figura 2), e ela produz um plano (linha 2) do tipo *PlanoExecucaoReal* (linha 3); tal plano possui as propriedades de *temCustoExecucao* (linha 4) e *temNumeroPaginasHipoteticas* (linha 5) com valores já previamente calculados no metadado do banco e por outra regra, respectivamente. Em seguida, ocorre a multiplicação das páginas por dois (linha 6) e o custo de execução é então, adicionado a esse resultado (linha 7). Caso o valor total seja maior do que zero (linha 8), a propriedade *temValorCustoEstimadoCriacao* recebe esse valor (linha 9). Com a definição do cálculo pela regra usando conceitos do próprio domínio do DBA, acredita-se que melhora o entendimento das regras usadas na decisão de sintonia-fina da ferramenta bem como facilita a definição de novas regras e heurísticas.

[2]https://www.tpc.org/tpch/

[3]`http://www.inf.puc-rio.br/~postgresql/conteudo/projeto4/webvowl/index.html`

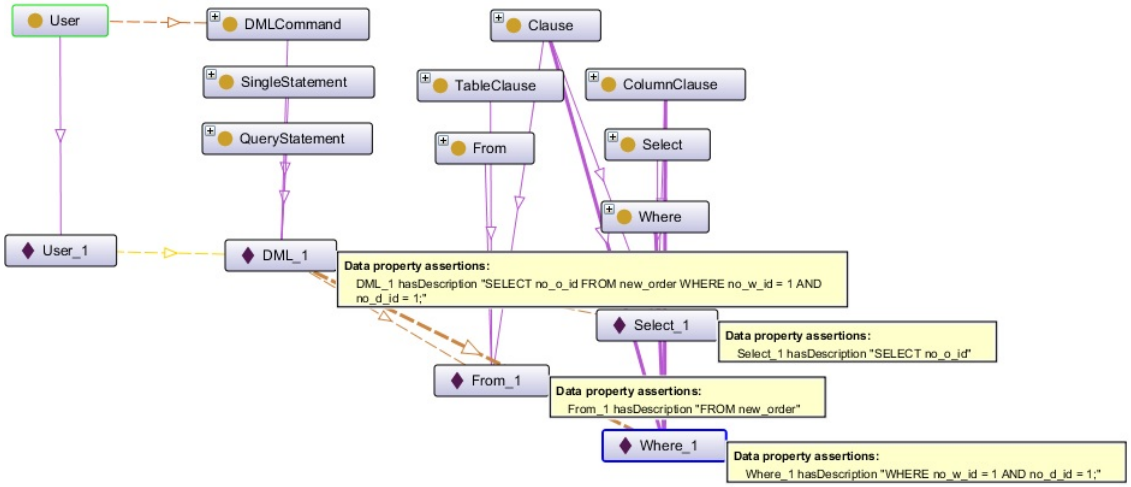


Figura 1. Fragmento da ontologia de domínio instanciada

```
1  VisaoMaterializadaHipotetica(?vmH) ∧
2     produz(?vmH, ?plan) ∧
3  PlanoExecucaoReal(?plan) ∧
4     temCustoExecucao(?plan, ?valorConsultar) ∧
5     temNumeroPaginasHipoteticas(?vmH, ?pagHipo) ∧
6     swrlb:multiply(?pagHipoMult, ?pagHipo, 2) ∧
7     swrlb:add(?total, ?pagHipoMult, ?valorConsultar) ∧
8     swrlb:greaterThan(?total, 0) →
9  temValorCustoEstimadoCriacao(?vmH, ?total)
```

Figura 2. Regra SWRL para cálculo do custo estimado de criação de uma VMH

3. Framework Outer-Tuning e arquitetura baseada em componentes

A arquitetura escolhida para o Outer-Tuning (Figura 3) é baseada em componentes. Devido ao caráter experimental da ferramenta e as múltiplas tecnologias envolvidas (e.g. SGBDs, máquinas de regras, ontologia, bibliotecas), decidiu-se que os componentes poderiam facilitar a comunicação entre as partes e deixar cada uma das fases de execução independentes. Assim, caso necessário, poderiam ser substituídos (ou mantidos) de forma compartimentalizada sem a propagação de *bugs*.

Como resultado da escolha de uma arquitetura baseada em componentes, decidiu-se que o Outer-Tuning seria desenvolvido como um *framework* de aplicação, que por definição é uma aplicação semicompleta, construída com uma coleção organizada de componentes de *software* reusáveis [Oliveira et al. 2011]. A escolha desse tipo de *framework* com uso de componentes, foi feita para possibilitar futura evolução para *software* orientado a serviços com baixo acoplamento entre as partes do *software*.

De acordo com o fluxo de execução proposto, o Outer-Tuning teve seus componentes definidos e especializados para cada etapa do ciclo. Na Figura 3, são enumerados os principais elementos da arquitetura proposta, descritos brevemente a seguir:

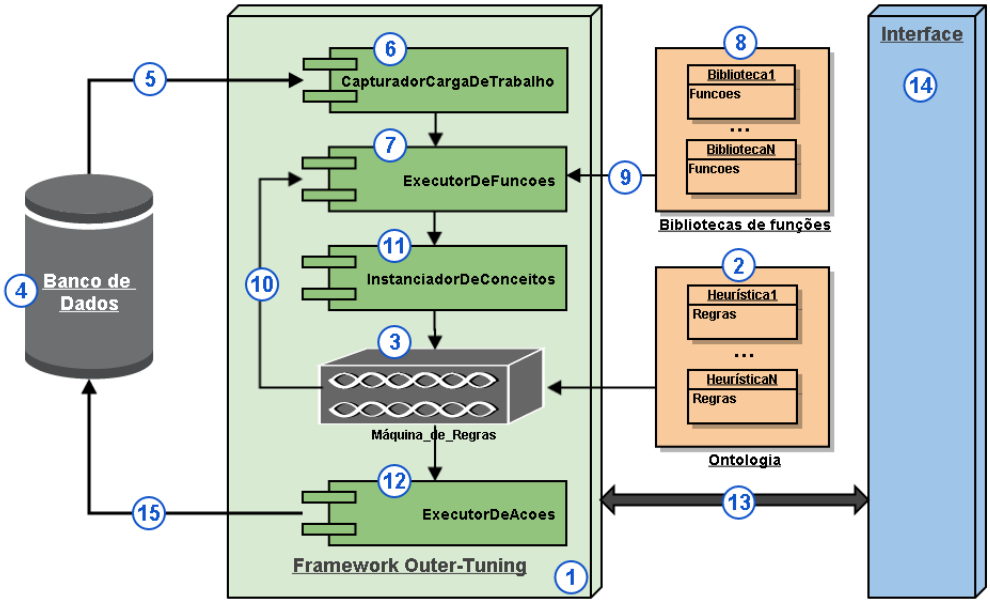


Figura 3. Arquitetura da Outertuning

(1) Base: Formada por uma biblioteca de funções para que os componentes possam compartilhar funções ordinárias e redundantes, além do acesso ao log compartilhado.

(2) Ontologia: A ontologia de aplicação contém a ontologia de domínio (conceitos instanciados pela carga de trabalho) e a ontologia de tarefas (heurísticas através de regras "se-então"). É a principal parte extensível do *framework*. As regras são definidas em uma linguagem declarativa (SWRL - Semantic Web Rules Language).

(3) Máquina de regras: Uma máquina de regras (ou motor de inferência) é o componente pelo qual as regras definidas na ontologia de tarefas são selecionadas e executadas. Para a implementação foi utilizada a máquina de regras Jess [4].

(4) Banco de dados: Qualquer banco de dados gerenciado por um SGBD possui sua comunicação com o *framework* (5) e (15) através dos *drivers* de conexão usados pelos componentes *CapturadorCargaDeTrabalho* (6) e *ExecutorDeFuncoes* (7).

(5) Carga de trabalho: A carga de trabalho capturada consiste em comandos SQL do tipo DML (*Data Manipulation Language* - Linguagem de manipulação de dados), os seus respectivos planos de execução e sua frequência.

(6) *CapturadorCargaDeTrabalho*: obtém a carga de trabalho com um intervalo de tempo pré-determinado pelo DBA para realizar a sintonia fina, através de driver JDBC [5].

(7) *ExecutorDeFuncoes*: Extrai informações da carga de trabalho e gera os indivíduos dos conceitos, que são pré-condições das heurísticas e que devem ser instanciados na máquina de regras para que haja inferência de ações de sintonia fina.

(8) Bibliotecas de funções: Sua função é aglutinar bibliotecas de código fonte compiladas, responsáveis por extrair conceitos da carga de trabalho. As bibliotecas são

[4]http://www.jessrules.com/ acesso em 25/05/2018
[5]http://www.oracle.com/ acesso em 25/05/2018

lidas e executadas em tempo de execução, sem intervenção no código fonte do *framework*.

(9) Comunicação entre bibliotecas e *ExecutorDeFuncoes*: realizada através de interface definida no *ExecutorDeFuncoes* que busca no repositório as funções desejadas.

(10) Conceitos pré-condições das heurísticas: O componente *ExecutorDeFuncoes* recebe da máquina de regras os conceitos que são pré-condições e a assinatura das funções contidas na biblioteca de funções que extraem conceitos.

(11) *InstanciadorDeConceitos*: Deve instanciar os indivíduos de pré-condições gerados pela execução das funções pelo *ExecutorDeFuncoes* na máquina de regras.

(12) *ExecutorDeAcoes*: Monitora a máquina de regras, captura as ações de sintonia fina inferidas e as executa no banco de dados de acordo com a escolha do DBA.

(13) Comunicação *framework* – interface: uso do padrão de projeto quadro-negro (*blackboard*)[Khosla et al. 2004], pela sua facilidade de implementação.

(14) Interface: Responsável pela interação com o DBA.

4. Outer-tuning na prática

Devido à limitação de espaço, vamos ilustrar brevemente o uso do Outer-Tuning. O vídeo em `http://www.inf.puc-rio.br/~postgresql/conteudo/projeto4/video/outertuning.mp4` mostra um possível uso da ferramenta.

Inicialmente, na Figura 4(A), o usuário do *framework* pode visualizar as heurísticas definidas na ontologia de tarefa e selecionar aquelas que deseja considerar para as futuras sugestões de sintonia fina. Posteriormente, o usuário informa para a ferramenta o modo que deseja trabalhar: semiautomático ou automático (sem intervenção humana). A ferramenta inicia a captura da carga de trabalho, em tempo real e apresenta, de forma gráfica, o momento em que o comando DML é executado no banco de dados e a sua duração, em segundos (Figura 4(B)). Caso o usuário queira detalhes maiores sobre a execução do comando, ele pode verificar mais abaixo na mesma tela (Figura 4(C)).

Caso o usuário queira acompanhar as ações de sintonia fina que estão sendo analisadas e sugeridas pela ferramenta, pode fazer isso através do menu *Tuning actions*. Nessa tela (Figura 4(D)), o usuário pode visualizar um gráfico com o cruzamento das informações de ganho esperado com a ação de sintonia fina (eixo x) e custo estimado de criação da estrutura de acesso (eixo y). O tamanho do círculo indicado no gráfico representa a quantidade de consultas SQL que a ação pode beneficiar. Quanto maior o tamanho do círculo, maior será o número de comandos beneficados pela ação de sintonia fina na carga de trabalho. O pop-up apresentado é um resumo sobre a ação de sintonia fina proposta com as seguintes informações: ganho esperado, custo de criação, tipo de ação (ex.: índice ou visão materializada) e número de comandos beneficiados pela ação.

As heurísticas que propõem VMs foram executadas e tiveram seus resultados avaliados e comparados através do Outer-Tuning. Para a geração da carga de trabalho durante os testes foi utilizado o *benchmark* TPC-H , propício para a avaliação de ferramentas de seleção de VMs por ser OLAP. Nota-se que a ferramenta apresenta tanto as avaliações positivas quanto as negativas. Algumas sugestões positivas foram implementadas e trouxeram benefícios conforme o esperado. Mais detalhes em [Oliveira 2015].

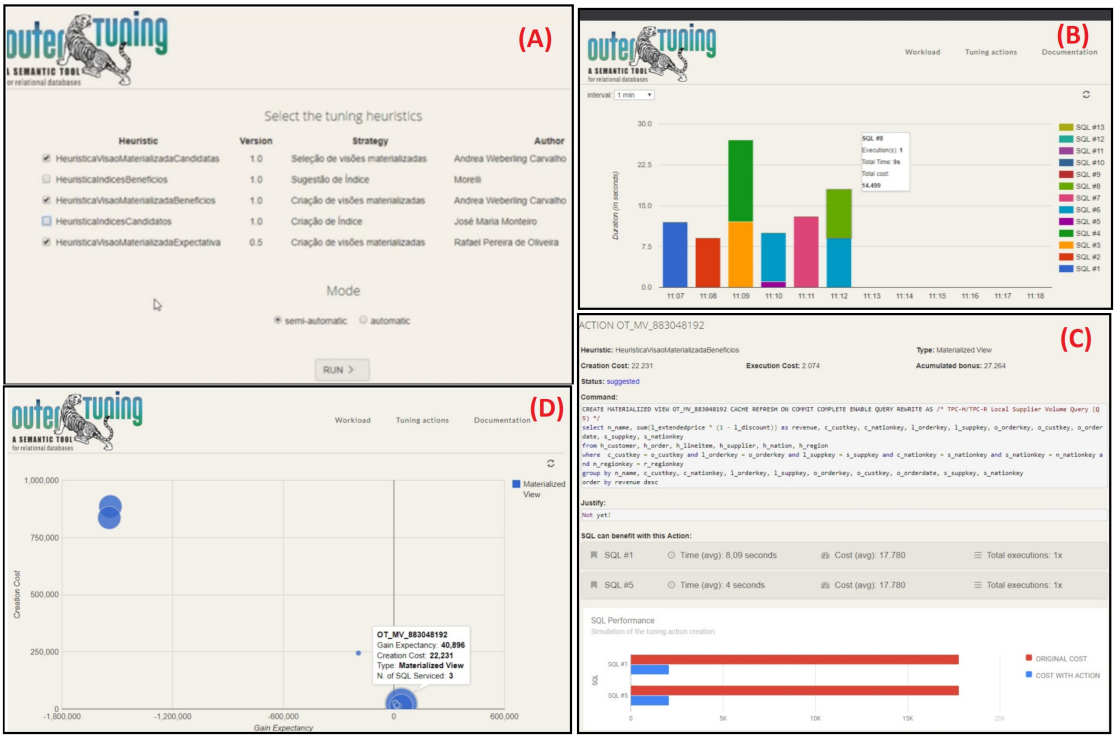


Figura 4. Amostra de Telas da Ferramenta Outer-Tuning

5. Comentários Finais

Apresentamos o *framework* Outer-Tuning para apoio à decisão sobre sintonia fina de bancos de dados relacionais. A ferramenta é apoiada por uma ontologia específica ao domínio que explicita conceitos e permite inferências. É possível visualizar as heurísticas instanciadas e o uso da sintonia fina de forma semiautomática.

References

[Almeida 2013] Almeida, A. C. B. d. (2013). *Framework para apoiar a sintonia fina de banco de dados*. PhD thesis, Depto. Informática - PUC-RIO.

[Goasdoué et al. 2011] Goasdoué, F., Karanasos, K., Leblay, J., and Manolescu, I. (2011). View selection in semantic web databases. *PVLDB*, 5(2):97–108.

[Khosla et al. 2004] Khosla, R., Ichalkaranje, N., and Jain, L. C. (2004). *Design of Intelligent Multi-Agent Systems: Human-Centredness, Architectures, Learning and Adaptation*. Studies in Fuzziness and Soft Computing. Springer.

[Khouri et al. 2012] Khouri, S., Bellatreche, L., Boukhari, I., and Bouarar, S. (2012). More investment in conceptual designers: Think about it! In *Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on*, pages 88–93. IEEE.

[Oliveira et al. 2011] Oliveira, J. L. D., Fernando, L., Loja, B., Larissa, S., Vicente, V., and Neto, G. (2011). Um Componente para Gerência de Processos de Negócio em Sistemas de Informação. *VII Simpósio Brasileiro de Sistemas de Informação*, pages 250–261.

[Oliveira 2015] Oliveira, R. P. d. (2015). Sintonia fina baseada em ontologias: o caso das visões materializadas. Master's thesis, Depto. Informática - PUC-RIO.

Anelim: Uma Ferramenta de Geração Automática de Dados para Banco de Dados Relacional em Ambientes de Testes

Angelo Brayner[1], **F. Ronald Araújo B.**[2], **José Maria Monteiro** [1]

[1] Departamento de Computação – Universidade Federal do Ceará (UFC)

[2]Departamento de Engenharia de Teleinformática – Universidade Federal do Ceará (UFC)

{brayner,monteiro}@dc.ufc.br, f.ronaldaraujo@gmail.com

Resumo. *Há uma crescente demanda pela geração de dados de testes, tanto para validar o projeto lógico e físico de bancos de dados, quanto para testar artefatos de software. Ademais, as aplicações atuais manipulam volumes de dados da ordem de magnitude de terabytes e até petabytes. Contudo, a criação manual de dados para tais cenários é inviável. Neste sentido, foi desenvolvida a ferramenta Anelim*[1]*, a qual tem por finalidade possibilitar a geração automática de grandes volumes de dados de testes. A partir da análise dos resultados obtidos, pode-se afirmar que, diferentemente de outras soluções já existentes, a Anelim gera uma massa de dados consistente, garantindo todas as restrições de integridade especificadas no esquema do banco de dados relacional.*

Abstract. *There is a growing demand for test data, both for validating the logical and physical design of databases, and for testing software artifacts. Nowadays, applications should handle very large databases. However, for such scenarios, creating test datasets in a manual way is not feasible. Thus, we proposed a tool, called Anelim*[1]*, which aims to enable the automatic creation of very large test datasets. From the analysis of the results, we can conclude that, unlike other already existing solutions, Anelim generates a mass of consistent data, ensuring all data integrity restrictions defined in a relational database schema.*

1. Introdução

Atualmente, as aplicações computacionais manipulam volumes de dados nunca antes imaginados. Para ilustrar este fato, pode-se destacar que no final de 2017 existiam 364 milhões de cartões de crédito em uso nos EUA [2]. Neste sentido, para que seja possível testar tais aplicações é imprescindível a existência de uma massa de dados de testes na mesma ordem de grandeza dos banco de dados reais (ou de produção). Desta forma, torna-se imperiosa a utilização de ferramentas que possibilitem a geração automática de conjuntos de dados de teste.

Existem inúmeras ferramentas que povoam automaticamente bancos de dados. Contudo, tais ferramentas não garantem as restrições de integridade especificadas no esquema do banco de dados, como, por exemplo a integridade referencial [Garcia-Molina et al. 2011]. Muitas destas ferramentas geram dados para tabelas específicas e não para o banco de dados como um todo. Na Seção 4.2, várias destas ferramentas serão analisadas e comparadas com a ferramenta proposta.

[1]https://youtu.be/eEpFBpOdNmQ

[2]https://www.creditcards.com/credit-card-news/ownership-statistics.php

Neste artigo será apresentada a ferramenta Anelim, para a geração automática de dados para banco de dados de testes. Estes dados produzidos a partir dos metadados existentes no esquema do banco de dados. Diferentemente das ferramentas existentes, a Anelim gera uma massa de dados consistente, onde as restrições de integridade do modelo relacional especificadas no esquema são garantidas.

Este artigo está estruturado da seguinte forma. A Seção 2 descreve a ferramenta proposta. Na Seção 3, são apresentados os tipos de dados suportados pela *Anelim*. Por sua vez, a Seção 4 apresenta resultados obtidos com a execução da *Anelim*, bem como, analisa ferramentas concorrentes da apresentada neste artigo. Por fim, a Seção 4.2 conclui este trabalho.

2. A *Anelim*

A ferramenta *Anelim* foi desenvolvida com o objetivo de gerar dados aleatórios para bancos de dados relacionais a partir do seu esquema lógico. Para tanto, a ferramenta deve apresentar as seguintes propriedades. Primeiramente, deve ter a capacidade de ler o esquema do banco de dados. Com esta propriedade pretende-se garantir as restrições de domínio (tipo de dados), chave (chave primária e chave candidata) e de integridade referencial especificadas nos metadados. Em segundo lugar, deve-se garantir ao usuário a possibilidade de especificar o volume de dados a ser gerado. Este volume refere-se ao banco de dados todo ou pode ser especificado para cada tabela. Por último, deve-se garantir ao usuário a opção de inserir ou não os dados automaticamente no banco de dados. Estas três propriedades, implementadas na ferramenta desenvolvida, introduziram um alto grau de complexidade à sua construção.

Com relação à garantia das restrições de integridade, a real dificuldade está em garantir a restrição de integridade referencial (IR). Isto se deve ao fato que a IR impõe uma ordem de inserção de tuplas nas tabelas, para que não seja violada a IR. Para ilustrar este problema, considere a modelagem conceitual entre os conjuntos de entidades Departamentos e Funcionários, apresentado na Figura 1. Observe que uma relação de cardinalidade um-para-muitos entre os dois conjuntos de entidades, onde Departamentos situa-se no lado *um* e Funcionários no lado *muitos*.

Desta forma, no esquema do banco de dados correspondente, o atributo *DepartamentoId* da tabela Funcionários é definido como chave estrangeira. Por este motivo, *DepartamentoId* deve possuir um valor que faz referência a um valor de chave primária na tabela Departamentos. Neste caso específico, o atributo *DepartamentoId* não pode assumir valor nulo, pois Funcionários apresenta participação total no relacionamento, ou seja todo funcionário deve estar lotado em um departamento. Portanto, não é possível inserir um funcionário sem informar em que departamento o mesmo está associado.

Em outras palavras, para gerar dados para o atributo *DepartamentoId*, a ferramenta precisa ler os valores gerados para a chave primária da tabela Departamentos. Caso contrário, a inserção de dados em Funcionários seria afetada, pois poderia haver muitos erros de violação da integridade referencial.

Para garantir todas as restrições de integridade do modelo relacional, a ferramenta proposta tem como entrada um arquivo de configuração, contendo os metadados do banco de dados. O formato padrão escolhido foi o *JavaScript Object Notation* (JSON). O ar-

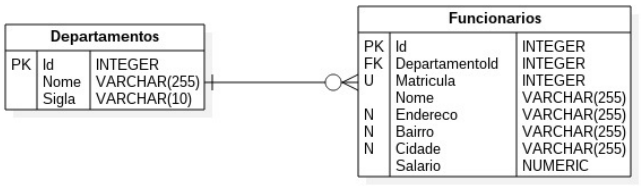


Figura 1. Entidades Departamentos e Funcionários.

quivo de configuração contém os metadados e parâmetros de geração de dados apresentados na Tabela 1. Adicionalmente, a *Anelim* solicita ao usuário os parâmetros apresentados na Tabela 2.

Tabela 1. Atributos do arquivo de configuração

Nome	Especificação	Obrigatório
tables	*Array* de objetos contendo as características de cada tabela.	Sim
number_inserts	Número de tuplas a ser inseridas em cada tabela.	Sim
tables → *name*	Nome da tabela.	Sim
fields	*Array* de objetos contendo as características de cada atributo da tabela.	Sim
fields → *name*	Nome do atributo.	Sim
primary_key	Informa se o atributo é chave primeira.	Não
type	Informa qual o tipo do atributo.	Sim
foreign_key	Informa se o atributo é chave estrangeira.	Não

3. Tipos de Dados Suportados

Atualmente, a *Anelim* suporta dois sistemas gerenciadores de bancos de dados (SGBDs), a saber: SQL Server e PostgreSQL. Cada um desses sistemas de bancos de dados apresenta um conjunto de diferentes tipos de dados. Todavia, oferecer suporte a todos esses tipos de dados específicos aumenta a complexidade do processo de geração de dados. Neste sentido, definiu-se um conjunto comum de tipos de dados suportados. O critério estabelecido para a escolha desses tipos foi a interseção entre estes conjuntos. A Tabela 3 mostra os tipos de dados que a *Anelim* pode gerar.

4. Experimentos e Análise Comparativa

4.1. Análise de Desempenho da *Anelim*

O principal objetivo desta análise foi averiguar o comportamento da *Anelim* no ato da geração e inserção dos dados nos dois SGBDs suportados. Para realizar o comparativo de desempenho, foi usado o esquema do banco de dados *Northwind*. Este banco de dados é composto de oito tabelas. O relacionamento entre estas tabelas é apresentado na Figura 2.

Por sua vez, a figura 3 mostra o gráfico dos tempos de execução para geração e inserção de dados no banco de dados, tanto no SQL Server, quanto no PostgreSQL.

Tabela 2. Atributos do arquivo de configuração

Nome	Valor Padrão	Funcionalidade
-f, --file	schema.json	Permite que o usuário informe o nome do arquivo contendo o schema do banco de dados.
-t, --target	mssql	Permite que o usuário informe a sintaxe que deverá ser utilizada na criação dos dados e/ou tabelas.
-d, –drop	false	Quando sinalizada com true, gera um script de “DROP TABLE” acima do script de inserção.
-c, --create	false	Quando sinalizado com true, gera um script de “CREATE TABLE” acima do script de inserção.
-i, --insert	false	Quando sinalizado com true, executa de forma transacional o script dentro do banco de dados.
--debug	false	Quando sinalizado com true, ativa o modo debug da ferramenta informando o seu passo-a-passo.

Tabela 3. Relação de tipos de dados aceitos

Tipo do dado	SQL Server	PostgreSQL
smallint	×	×
integer	×	×
bigint	×	×
decimal	×	×
real	×	×
serial		×
money	×	×
varchar	×	×
date	×	×
time with time zone		×
boolean		×
uuid		×
bit	×	×

Observe que cada tabela do Northwind foi povoada com uma quantidade de 10, 100 e 1000 tuplas.

Vale destacar que os resultados apresentados na Figura 3 foram obtidos com a execução de comandos de *insert* convencional. Em outras palavras, para cada *insert*, o SGBD verificava a garantia das restrições de integridade, como, por exemplo, restrição de chave, de identidade e integridade referencial. Atualmente, estamos implementando estratégias mais eficientes para inserção de grandes volumes de dados. Para o SQL Server, por exemplo, estamos implementado a estratégia de *bulk insert*.

4.2. Análise Comparativa

Esta seção apresenta uma análise comparativa entre a *Anelim* e as principais ferramentas relacionadas: DataFiller, Database Test Data, GenerateData, DGMaster e Mockaroo. Os critérios utilizados nesta comparação foram: i) a observâncias das restrições de integridade definidas no esquema lógico e ii) os SGBDs suportados.

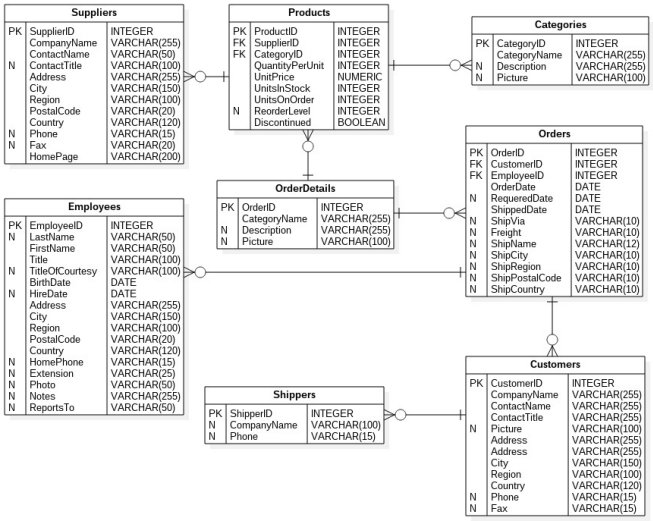


Figura 2. Banco de dados Northwind.

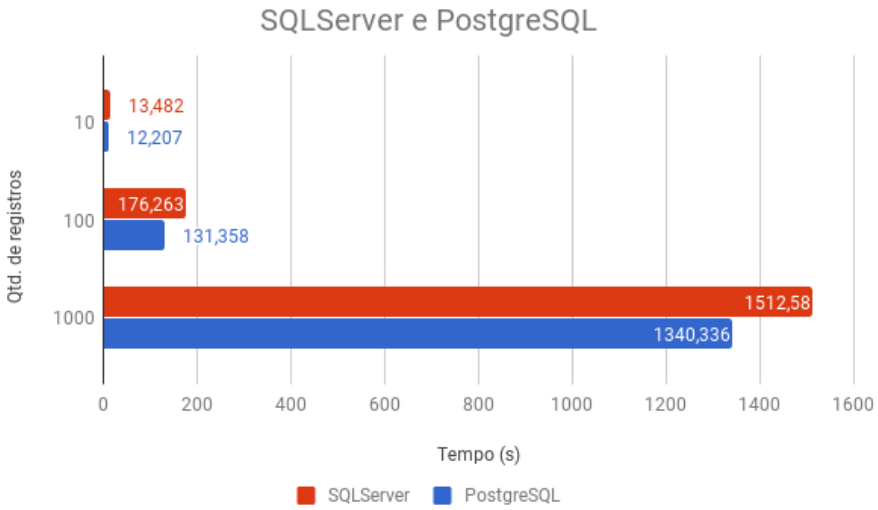


Figura 3. Desempenho da ferramenta.

A ferramenta DataFiller [DataFiller 2014] implementa uma estratégia de inserção por tabela. Assim, é possível informar a quantidade e probabilidade de geração de dados para cada atributo da tabela. Contudo, esta ferramenta não garante a restrição de integridade referencial de forma satisfatória. Tal característica, induz a muitos erros de violação de integridade referencial durante a inclusão dos dados gerados. Consequentemente, a DataFiller não garante que a quantidade de dados solicitados pelo usuário seja gerada de forma satisfatória. Por fim, esta ferramenta só consegue gerar e inserir dados para o PostgreSQL.

As ferramentas GenerateData [GenerateData 2017] e Database Test Data [Test Data Generator 2018] não permitem a inserção automática no banco de dados, apenas geram arquivos contendo os dados gerados. Consequentemente, não garantem as restrições de integridade do modelo relacional.

Por sua vez, a DGMaster [DGMaster 2017] apresenta suporte a vários tipos de

dados, gerando-os em diversos formatos (por exemplo, texto e XML). Contudo, além de apresentar uma interface confusa e de difícil utilização, apresenta instabilidade operacional. Finalmente, com a capacidade de gerar grandes volumes de dados em vários modelos de dados diferentes (relacional, XML, planilhas Excel, entre outros), a ferramenta [Mockaroo 2017] não apresenta suporte para povoar o banco de dados de forma automática.

A Tabela 4 ilustra o resultado da análise comparativa realizada. Pode-se perceber, analisando a Tabela 4, que *Anelim* é a única ferramenta a respeitar as restrições de integridade definidas no esquema lógico. Ademais, a *Anelim* fornece suporte para dois diferentes SGBDs.

Tabela 4. Análise comparativa entre ferramentas de geração de dados

Ferramenta	*Respeita as RIs*	*SGBDs Suportados*
Anelim	Sim	2
Database Test Data	Não	0
DataFiller	Não	1
GenerateData	Não	0
DGMaster	Sim	0
Mockaroo	Não	0

5. Considerações Finais

Neste trabalho, apresentou-se uma ferramenta, denominada Anelim, capaz de gerar automaticamente dados de teste em banco de dados relacionais. Atualmente, a Anelim é capaz de povoar bancos de dados em dois SGBDs, o SQLServer e o PostgreSQL. A partir da análise dos resultados obtidos, pode-se inferir que a *Anelim* apresenta diversas vantagens em relação às ferramentas concorrentes avaliadas neste trabalho. Atualmente, estão sendo implementadas estratégias para inserção de grandes volumes de dados e suporte para inserção de dados no MySQL.

6. Referências

Referências

DataFiller (2014). Generate random data from database schema. `https://www.cri.ensmp.fr/people/coelho/datafiller.html`. Abril.

DGMaster (2017). Data generator: simple, free, extensible. `http://dgmaster.sourceforge.net/`. Maio.

Garcia-Molina, H., Ullman, J., and Widom, J. (2011). *Database Systems: The Complete Book*. Pearson Education.

GenerateData (2017). Script generate random data from a database schema. `http://www.generatedata.com/`. Abril.

Mockaroo (2017). Random Data Generator and API Mocking Tool. `https://mockaroo.com/`. Maio.

Test Data Generator, D. (2018). Fill your database with random test data. `http://www.databasetestdata.com/`. Abril.

MobileECG: Uma Ferramenta para Publicação e Integração de Dados de Sinais ECG

Tibet Teixeira[1], Francisco San Diego Castilho[1], Daniel Rodrigues[1], Douglas Torquato[1], João Paulo Madeiro[2], José Maria Monteiro[1], Angelo Brayner[1], Vânia Vidal[1], Narciso Arruda[1], Tiago Vinuto[1]

[1]MDCC – Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brasil

[2]IEDS – UNILAB
Redenção, CE – Brasil

{tibet,sandiego,daniel,douglas}@lia.ufc.br

{monteiro,brayner,vvidal,narciso,tiagosv}@lia.ufc.br

jpaulo.vale@unilab.edu.br

Abstract. *The ECG signal acquisition is a simple and relatively inexpensive diagnostic tool, important for monitoring people suffering from a plethora of heart diseases. Extracting features from the ECG signal allows to comprehend details related to cardiac activity, which may present subtle changes, regular or irregular patterns, and so on. At this sense, we present here a tool named MobileECG[1], which provides signal acquisition, ECG feature extraction, besides ECG data integration and publication using Linked Data. Thus, MobileECG supplies a public knowledge base, which may be used to support complex queries, run mining algorithms and to yield collaboration among experts.*

Resumo. *A aquisição do sinal ECG consiste numa técnica relativamente simples, não-invasiva e de baixo custo, que permite o monitoramento de pacientes acometidos por uma diversidade de doenças cardíacas. A extração de informações ou parâmetros do sinal ECG possibilita a análise e compreensão da atividade cardíaca, a qual pode apresentar padrões regulares ou irregulares e alterações súbitas. Neste cenário, é apresentada a ferramenta MobileECG, a qual realiza a aquisição do sinal ECG, a extração de parâmetros do sinal, a extração de dados do sinal ECG, além da integração e publicação desses dados, seguindo os principais padrões para o compartilhamento de dados abertos na Web. Assim, a ferramenta MobileECG fornece uma base de conhecimento pública, que pode ser usada para dar suporte a consultas complexas e algoritmos de mineração, além de possibilitar a colaboração entre especialistas.*

1. Introdução

O sinal ECG é o registro das diferenças de potencial produzidas pela atividade elétrica das células cardíacas. O corpo humano por si só atua como um grande condutor de corrente

[1]Um vídeo de demonstração da ferramenta MobileECG pode ser encontrado em: http://tiny.cc/mobileecg

elétrica, e quaisquer dois pontos na superfície podem ser conectados por eletrodos para registrar um ECG ou monitorar o ritmo do coração. 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.

Devido ao aumento contínuo dos custos com tratamentos, acompanhamentos médicos e cuidados com a saúde, as doenças crônicas não transmissíveis (DCNT), incluindo-se as doenças cardíacas, derrame, câncer, diabetes e doença pulmonar crônica, são coletivamente responsáveis por quase 70% de todas as mortes no mundo, de acordo com a Organização Mundial de Saúde (OMS). Adicionalmente, considerando também que a população mundial está envelhecendo, firma-se uma necessidade significativa de monitorar o estado de saúde de um paciente enquanto ele está em seu ambiente cotidiano. Nesse contexto, uma ampla variedade de protótipos de sistemas desenvolvidos visa fornecer informações em tempo real sobre a condição de saúde de um indivíduo, seja para o próprio usuário, para um centro médico ou diretamente para um cardiologista.

Os denominados sistemas vestíveis para monitoramento da saúde compreendem vários tipos de sensores em miniatura, vestíveis ou mesmo implantáveis, capazes de medir de um a três derivações de ECG, entre outros sinais fisiológicos, e posteriormente transmitir a informação extraída por meio de um link, possivelmente sem fio, para um dispositivo microcontrolador. Esse dispositivo pode transmitir o sinal condicionado e digitalizado para um *smartphone*, um computador ou um banco de dados colaborativo. Em seguida, algoritmos baseados em técnicas de processamento digital de sinais, segmentação de formas de onda e reconhecimento de padrões de doenças cardíacas podem ser executados em uma variedade de plataformas microprocessadas. Portanto, uma solução completa de monitoramento de sinais vitais engloba uma ampla variedade de componentes: sensores, materiais vestíveis, fontes de alimentação, comunicação sem fio, unidades de processamento, interface para o usuário, software e algoritmos para extração de parâmetros, estruturação e publicação dos dados [Pantelopoulos and Bourbakis 2010].

2. A Ferramenta MobileECG

Este trabalho apresenta uma ferramenta, MobileECG, a qual possibilita a aquisição do sinal ECG, a extração de parâmetros das formas de onda, a extração de dados dos sinais ECG, além da integração e publicação de dados de ECG, seguindo os principais padrões para o compartilhamento de dados abertos. Desta forma, a ferramenta MobileECG fornece uma base de conhecimento pública, que pode ser usada para dar suporte a consultas com diferentes níveis de complexidade, executar algoritmos de mineração de dados e aprendizagem de máquina, além de possibilitar a colaboração entre especialistas.

A Figura 1 ilustra a arquitetura da ferramenta MobileECG, a qual compreende os seguintes módulos: módulo de aquisição, módulo de aplicação no *smartphone* do paciente, módulo de processamento do sinal ECG e extração de parâmetros, módulo de extração de dados e módulo de integração e publicação de dados. O módulo de aquisição foi implementado utilizando-se um biossensor e um arduino. O módulo de aplicação foi implementado usando-se a plataforma Android e é executado no *smartphone* do paciente. Os módulos de processamento do sinal ECG e extração de parâmetros, de extração de

dados e de integração e publicação de dados foram implementados utilizando-se a tecnologia de Serviços Web e a linguagem Java. Esses três módulos são executados em uma nuvem computacional e podem ser acessados de forma independente de localização. Todos os módulos serão descritos a seguir.

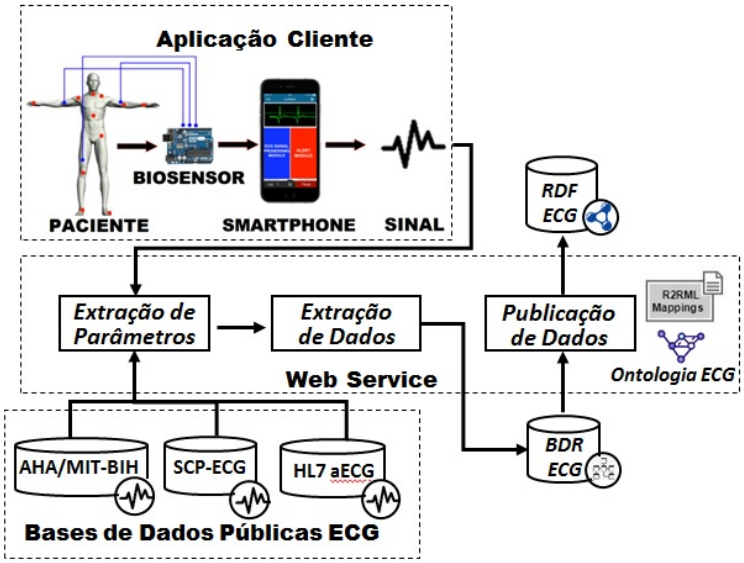


Figura 1. Arquitetura da ferramenta MobileECG

2.1. Módulo de Aquisição

O módulo de aquisição é formado por um biossensor acoplado em um microcontrolador Arduíno. A captação do sinal ECG é realizada pelo biossensor, que mede e amplifica os potenciais elétricos derivados da atividade elétrica do coração, e, em seguida, transmite o sinal analógico para o Arduíno. O biossensor utilizado, modelo *SHIELD-EKG-EMG*, é um hardware de código aberto que permite comunicação direta (*SHIELD*) com um microcontrolador Arduíno, o qual, por sua vez, recebe o sinal analógico, realiza a conversão Analógica/Digital deste sinal e, em seguida, transmite o sinal digital para o *smartphone* do paciente utilizando um módulo (*SHIELD*) de comunicação *bluetooth*.

2.2. Módulo de Aplicação no Smartphone do Paciente

O módulo de aplicação consiste em um aplicativo (*App*) desenvolvido utilizando-se a plataforma Android que é executado no *Smartphone* do Paciente. Esta aplicação possui uma interface gráfica simplificada que permite realizar algumas atividades básicas, tais como: iniciar e finalizar o processo de aquisição do ECG, além de possibilitar a visualização do sinal recebido. O aplicativo permite ainda que o paciente registre eventos específicos, considerando qualquer queixa sobre seu estado de saúde: dor torácica, falta de ar, cefaleia, tontura, palpitações e batimento cardíaco acelerado, entre outros. Por fim, o aplicativo transmite o sinal recebido para o módulo de processamento do sinal e extração de parâmetros, por meio de um serviço Web.

2.3. Módulo de Processamento do Sinal e Extração de Parâmetros

Este módulo consiste em um serviço Web implementado em linguagem Java, o qual contém um conjunto de algoritmos de processamento digital de sinais, tais como: fil-

tragem para eliminação de ruído e interferências, aplicação de transformada *Wavelet* (análise tempo-frequência) para realce seletivo do complexo QRS e das ondas P e T, detecção dos picos, delineamento das formas de onda e extração de parâmetros. Como resultado desta etapa, os seguintes parâmetros são extraídos: amplitude e duração de cada complexo QRS, intervalos entre batimentos, amplitude e duração de cada onda P e de cada onda T e intervalos entre as diferentes formas de onda [Madeiro et al. 2012, Madeiro et al. 2013, do Vale Madeiro et al. 2017]. Vale destacar que este módulo recebe como entrada um sinal ECG, o qual pode ter sido originado de um biossensor, ou seja, do módulo de aquisição, ou de bases públicas previamente existentes (como por exemplo, AHA/MIT-BIH, SCP-ECG e HL7 aECG).

2.4. Módulo de Extração dos Dados

Este módulo consiste em um serviço Web implementado em linguagem Java que recebe como entrada uma matriz contendo os parâmetros extraídos do sinal. As informações contidas nesta matriz são processadas, e um conjunto de dados sobre o sinal ECG é extraído e armazenado em banco de dados relacional. A abordagem de extrair e armazenar os dados do sinal ECG em um banco de dados relacional foi adotada com o objetivo de simplificar o processo de publicação dos dados, uma vez que já existem ferramentas que possibilitam a criação de *dumps* dos dados relacionais em formato RDF e a disponibilização desses *dumps* em *triplestore* RDF, de forma semiautomática.

2.5. Módulo de Integração e Publicação dos Dados

O módulo de integração e publicação de dados consiste em um serviço Web implementado em linguagem Java que acessa o banco de dados relacional contendo as informações extraídas dos sinais ECG e exporta os dados relacionais para o formato RDF. Dividimos esse processo em duas etapas. Na primeira etapa, cria-se um *dump* dos dados relacionais em formato RDF. Para executar esta etapa, utilizamos a ferramenta *D2RQ* [2] junto com os mapeamentos na linguagem R2ML [3], os quais relacionam o esquema do banco de dados relacional com o vocabulário da ontologia adotada para representar os sinais ECG. Na segunda etapa, os dados em formato RDF presentes no *dump* gerado anteriormente são materializados em um *triplestore* RDF, mais especificamente no Virtuoso [4], de forma semiautomática. O Virtuoso disponibiliza um *SPARQL endpoint* que possibilita realizar consultas semânticas.

3. Estudo de Caso

Realizar consultas sobre dados de sinais ECG é uma tarefa bastante complexa, uma vez que tais dados são armazenados em formatos heterogêneos e não estruturados. Por outro lado, a publicação de dados de sinais ECG seguindo-se as premissas de *Linked Data* possibilita a realização de consultas semânticas. Para demonstrar este fato, apresentamos duas perguntas que podemos responder seguindo-se as referidas premissas.

3.1. Consultas SPARQL

Consulta Sparql 01: Quais pacientes do sexo masculino com mais de 60 anos tiveram algum batimento acelerado (acima de 100 batimentos por minuto) no ECG?

[2]http://d2rq.org/

[3]https://www.w3.org/TR/r2rml/

[4]https://virtuoso.openlinksw.com/rdf/

```
prefix ecga:  <http://www.arida.ufc.br/ecg> .
prefix health:  <https://health-lifesci.schema.org/> .
prefix ecg:  <http://nemo.inf.ufes.br/biomedicine/ecg.html> .
SELECT ?paciente
WHERE {?paciente a health:Patient ; health:age ?idade ;
           health:gerner ?sexo ; ecga:hasECG ?ecg .
      ?ciclo ecgo:part_of ?ecg .
      ?ciclo ecgo:hasBeat ?bat .
      ?bat a ecgo:FastBeat.
FILTER(?age >60 && ?sexo = "male")}
```

Figura 2. Consulta Sparql 01.

Consulta Sparql 02: Quais pacientes tomaram o medicamento *Aldomet* e apresentaram algum batimento cardíaco lento (abaixo de 60 batimentos por minuto) no ECG?

```
prefix ecga:  <http://www.arida.ufc.br/ecg> .
prefix health:  <https://health-lifesci.schema.org/> .
prefix ecg:  <http://nemo.inf.ufes.br/biomedicine/ecg.html> .
SELECT ?paciente
WHERE {?paciente a health:Patient ; health:drug ?med .
      ?med rdfs:label ?nomemed .
      ?paciente ecgo:hasECG ?ecg .
      ?ciclo ecgo:part_of ?ecg ; ecgo:hasBeat ?bat .
      ?bat a ecgo:SlowBeat.
FILTER (?nomemed = "Aldomet")}
```

Figura 3. Consulta Sparql 02.

4. Trabalhos Relacionados

Em [Ngo and Veeravalli 2014], DuyHoa Ngo et al. propõem uma plataforma baseada nas tecnologias da Web Semântica que permitem o armazenamento de parâmetros extraídos do sinal ECG em uma base de dados seguindo-se os padrões de *Linked Data*. Os autores ressaltam ainda que a plataforma proposta é parte de um sistema de atenção à saúde baseado em armazenamento de dados em nuvem, o qual também captura os sinais ECG e outros sinais vitais através de biossensores localizados na superfície do corpo do paciente. Em [Trigo et al. 2012], Jesús Daniel Trigo et al. propõem o projeto e o desenvolvimento de um Sistema Integrado de Informações de Atenção à Saúde (IHIS - *Integrated Healtcare Information System*) para o gerenciamento de interoperabilidade de diferentes formatos de registros digitais de sinais ECG. Os autores propõem uma combinação de diferentes formatos de armazenamento de registros de ECG e uma arquitetura de software baseada em sistemas de informação empresarial (EIS) que permita uma padronização de interoperabilidade entre padrões e um gerenciamento homogêneo de diferentes formatos/padrões de armazenamento de sinais ECG. Em [Khumrin and Chumpoo 2016], Piyapong Khumrin e Pitupoom Chumpoo propõem uma abordagem visando à integração de dados eletrocardiográficos originados de diferentes formatos e a implementação de um sistema integrado de informações de dados eletrocardiográficos no contexto do sistema de informação do Hospital Maharaj Nakorn Chiang Mai (Tailândia). O sistema proposto integra diferentes formatos de dados de ECG utilizando-se um software desenvolvido em linguagem Java,

cuja interface funciona como um *Middleware* no processo de integração. Os autores aplicam um formato de dados de referência baseado em estruturação de classes definidas no ambiente Java para mapeamento de diferentes formatos de dados de ECG.

5. Conclusões e Trabalhos Futuros

Neste trabalho, apresentamos uma ferramenta, denominada MobileECG. Podemos elencar diversos diferenciais entre a MobileECG e os trabalhos anteriores. A ferramenta MobileECG compõe uma solução completa de monitoramento de atividade cardíaca, compreendendo desde a etapa de aquisição do sinal, através de biossensores, conversão Analógica/Digital por meio do Arduíno, transmissão do sinal digital utilizando-se um aplicativo desenvolvido em Android disponibilizado em um *smartphone*, além da extração dos parâmetros do sinal, da extração dos dados acerca do sinal, integração e publicação de dados usando-se os padrões de *Linked Data*, tarefas essas realizadas por meio de serviços Web. Por fim, a ferramenta MobileECG fornece uma base de conhecimento pública, que pode ser usada para dar suporte a consultas complexas, executar algoritmos de mineração, além de possibilitar a colaboração entre especialistas. Como trabalhos futuros, iremos inserir na MobileECG processos de anonimização dos dados dos pacientes, visando à proteção das informações individuais, compactação dos dados dos sinais ECG, bem como investigar e implementar algoritmos de aprendizado de máquina para reconhecimento e/ou predição de padrões de doenças cardíacas e outros eventos adversos.

Referências

do Vale Madeiro, J. P., dos Santos, E. M. B. E., Cortez, P. C., da Silva Felix, J. H., and Schlindwein, F. S. (2017). Evaluating gaussian and rayleigh-based mathematical models for t and p-waves in ecg. *IEEE Latin America Transactions*, 15(5):843–853.

Khumrin, P. and Chumpoo, P. (2016). Implementation of integrated heterogeneous electronic electrocardiography data into maharaj nakorn chiang mai hospital information system. *Health informatics journal*, 22(1):34–45.

Madeiro, J. P., Cortez, P. C., Marques, J. A., Seisdedos, C. R., and Sobrinho, C. R. (2012). An innovative approach of qrs segmentation based on first-derivative, hilbert and wavelet transforms. *Medical engineering & physics*, 34(9):1236–1246.

Madeiro, J. P., Nicolson, W. B., Cortez, P. C., Marques, J. A., Vázquez-Seisdedos, C. R., Elangovan, N., Ng, G. A., and Schlindwein, F. S. (2013). New approach for t-wave peak detection and t-wave end location in 12-lead paced ecg signals based on a mathematical model. *Medical engineering & physics*, 35(8):1105–1115.

Ngo, D. and Veeravalli, B. (2014). Applied semantic technologies in ecg interpretation and cardiovascular diagnosis. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 17–24. IEEE.

Pantelopoulos, A. and Bourbakis, N. G. (2010). A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12.

Trigo, J. D., Martínez, I., Alesanco, A., Kollmann, A., Escayola, J., Hayn, D., Schreier, G., and García, J. (2012). An integrated healthcare information system for end-to-end standardized exchange and homogeneous management of digital ecg formats. *IEEE Transactions on Information Technology in Biomedicine*, 16(4):518–529.

SBBD 2018

Workshop on Thesis and Dissertations in Databases

Combining meta-heuristics and linear programming to address ontology meta-matching problem

Nicolas Ferranti[1], **Stênio Sã Rosário Furtado Soares**[2], **Jairo F. De Souza**[1]

[1] Programa de Pós-Graduação em Ciência da Computação
Federal University of Juiz de Fora (UFJF) 36.036-900 – Juiz de Fora, MG – Brazil

[2]Department of Computer Science – Federal University of Juiz de Fora (UFJF)
36.036-900 – Juiz de Fora, MG – Brazil

{nicolas1, ssoares}@ice.ufjf.br, jairo.souza@ufjf.edu.br

Abstract. *Every year, several new ontology matchers are proposed in the literature, each one using a different heuristic, which implies in different performances according to the characteristics of the ontologies. An ontology meta-matcher consists of an algorithm that combines several approaches in order to obtain better results in different scenarios. To achieve this goal, it is necessary to define a criterion for the use of matchers. In this work, a study is proposed exploring approaches of distinct areas of computing, like meta-heuristics and linear programming for the construction of an ontology meta-matcher.*

Resumo. *Todo ano, diversos novos alinhadores de ontologias são propostos na literatura, cada um utilizando uma heurística diferente, o que implica em desempenhos distintos de acordo com as características das ontologias. Um meta-alinhador consiste de um algoritmo que combina diversas abordagens a fim de obter melhores resultados em diferentes cenários. Para atingir esse objetivo, é necessária a definição de um critério para melhor uso de alinhadores. Neste trabalho, é proposto um estudo explorando abordagens de areas distintas da computação, como meta-heurísticas e programação linear para a construção de um meta-alinhador de ontologias.*

The student entered the master's program in March 2018 with a forecast of defense for December 2019
The master's program does not have a qualification examination.

1. Introduction

Ontologies are built by people with distinct levels of expertise and domain view. Therefore, concepts that describe the same type of object may be represented in different ways, both in syntax terms and structure of relations, generating a problem of heterogeneity in data semantics. To solve problems of heterogeneity, a way is to unambiguously specify the vocabularies underlying information systems [Farinelli and Almeida 2014].

The problem to be solved consists of defining relationships between concepts of the input ontologies, making the structures compatible to represent the union of the data sets in a new model. The alignment of ontologies, as it is called, is a complex problem and its characteristics allow it to be approached by several computational techniques. Due to the high heterogeneity of the ontologies, there is no technique that stands out from the others in all aspects [Xue and Tang 2017]. Therefore, meta-matching approaches can be used in this scenario. A meta-matcher combines several alignment techniques in order to explore various aspects of heterogeneity to avoid alignment performance being restricted to some ontology characteristics. The literature suggests that the best results found in the meta-matching of ontologies are associated to the use of evolutionary algorithms due to their adaptability and, consequently, adequacy of the use of each technique [Souza 2012, Shi and Eberhart 1998, Xue and Tang 2017].

The use of meta-heuristics in meta-matchers is justified by the size of the search space of the problem, which discourages the use of exhaustive approaches due to the processing time demanded. In addition, the literature shows that population approaches present themselves as good alternatives in solving the problem. Although the use of Genetic Algorithms is more frequent between approaches, other bio-inspired approaches constitute a promising field to be explored in solving the problem. In the last decade, many meta-heuristics have been proposed and are still little explored in the database area [Sorensen et al. 2017].

Among the bio-inspired population approaches, the prey-predator algorithm (PPA) has characteristics that are adequate for the meta-matching problem, since it allows promising regions of the solution space to be exploited by a set of agents (solutions), pressured to runaway from regions that are not attractive in terms of the value of the objective function, while allowing the exploration of new regions by assigning to these agents a pseudo random behavior in the definition of their displacement.

The objective of this work is to deal with the ontology meta-matching problem through an adaptation of the prey-predator algorithm (PPA) and to show its applicability to this problem. To analyze the behavior of the initial solution, the benchmark provided by the OAEI (Ontology Alignment Evaluation Initiative) was used. It was observed that PPA is efficient and effective when parametrizing the alignment techniques in order to obtain a solution that is close to the optimum in polynomial time. Thus, the work shows that PPA can be better exploited in the ontology alignment scenario.

2. Related Work

The problem of meta-matching of ontologies is a relatively recent problem and still has several characteristics to be explored, although the known proposals have presented good results. In this section we present related works that use meta-heuristics to deal with the alignment of ontologies, highlighting their main characteristics and contributions.

The use of meta-heuristics has been explored to solve the meta-alignment problem, as seen in [Souza 2012, Xue and Tang 2017, Bock and Hettenhausen 2012]. In GNoSIS+ [Souza 2012], a genetic algorithm is used to parameterize a set of predefined aligners. Algorithm learning is based on a set of reference alignments defined by an ontology engineer input. The premise is that some relationships can be easily pointed out, so AG calibrates the system functions based on the reference in order to prepare it for a real application situation. It is interesting to highlight the representation of the problem by GNoSIS+. Considering $\Xi = \{F_1, F_2, ..., F_n\}$ a set of alignment functions, each chromosome has *n* genes($|\Xi| = n$) and each gene represents a real value $w \in [0, 1]$ representing the weight to be applied to each function. The goal is to minimize the difference between the value found and the value defined by the ontology engineer for a particular relationship. [Xue and Tang 2017] also employs an evolutionary algorithm, however the objective function is to maximize the value of the harmonic mean of f-measure. The f-measure is a metric that takes into account the precision and recall rates between the mappings obtained by the algorithm with those that were expected. The objective function of each work guides its algorithms to different paths. For the [Xue and Tang 2017] approach, it is necessary to evaluate each item of the result obtained at each iteration with the reference base, resulting in a computational cost greater than just comparing the result obtained with the confidence value defined by the engineer, as is done in [Souza 2012].

MapPSO [Bock and Hettenhausen 2012] is a solution that employs the particle swarm optimization to deal with the meta-matching problem. Particle swarm is a natural-inspired technique based on the social behavior of individuals, such as the flock of birds to find a place with food enough [Shi and Eberhart 1998]. The approach aims to find only equivalence type relations (1:1) and uses a predefined matcher that implements a distance function. The distance function defines a level of similarity for a given pair of concepts. It is important to note that MapPSO does not calibrate a set of alignment functions because it uses only one. However, it can be considered a meta-matching approach by seeking optimal matching by making use of predefined matchers. The representation of the individual differs from previous work. In this approach, each solution is represented as a candidate match. Suppose that $\vec{X_p}$ represents an alignment of two ontologies consisting of k = 5 matches (c). A particle is represented by $\vec{X_p} = \left\{c_{(p,1)}, c_{(p,2)}, c_{(p,3)}, c_{(p,4)}, c_{(p,5)}\right\}$ where each $c_{(p,i)}$ indicates a confidence value for the relationship (p, i).

The literature shows that evolutionary approaches have good results when applied in the ontology matching [Otero-Cerdeira et al. 2015], which allows to foment that the use of the prey-predator meta-heuristic has the potential to construct effective results for the problem.

The definition of the individual's representation impacts on how the effort of the approach can be reproduced. Once the representation is based on the set of weights, the found parameters can be stored and retrieved without much effort, whereas the representation based on the set of candidate alignments requires that the whole process be executed again. Therefore, the weights of the sets of weights have a better contribution to the construction of a more generic meta-aligner.

3. Proposed Solution

Ontologies matching problem allows several approaches to be used in the development of algorithms to the construction of solutions. This work intends to explore the variety of characteristics, trying the combination of approaches present in the literature in search of better solutions. Thus, we selected a set of experiments to be carried out based on the studies found in literature:

1. To explore the prey-predator meta-heuristic defined in [Tilahun and Ong 2015] as a means of solving the linear system constructed by [Souza 2012], as a new ontology meta-matcher.
2. To investigate the adaptation of the prey movement presented by [Tilahun and Ong 2015] to the problem. The discretization of the solution space is already used by other authors and can contribute to reach better solutions and reduce the processing time of the solution
3. Evaluate the construction time and quality impacts of the solutions when working with a multi-objective function, by combining metrics such as precision and recall in conjunction with the linear system solution
4. Use linear programming techniques like the Simplex algorithm to maximize the objective function of the problem

Initial efforts of this work turns to the first item. In a first step, the behavior was reproduced and used to solve the linear system constructed by [Souza 2012] using as an objective function the minimization of the differences found. Each year OAEI[1] provides a set of instances for ontology matching testing, these instances are used by meta-matcher developers to demonstrate the adaptive capacity of the algorithm.

Preliminary results indicate precision and recall rates reaching on average, 90%, a difference of 6% less than the work of [Souza 2012]. The prey-predator of [Tilahun and Ong 2015] as it was proposed, allows a solution to walk at any point in a continuous solution space, making the search space be the whole set of possible solutions, the next steps of the stage of using the prey-predator focus on the customization of meta-heuristic operators so that it can walk in a discrete way, described in item 2, the discretization of the solution space is employed by other authors in other meta-heuristics and the expectations are related to a better behavior of the algorithm, achieving better results in a shorter time frame.

[Souza 2012, Xue and Tang 2017] model the problem by constructing a linear system in order to parameterize a set of matchers, although the works are different, they have many similar characteristics, which allows to experiment the behavior of an approach that combines both using a multi-objective function, which take into account the solution of the linear system and the maximization of f-measure for a predetermined set of training. Once a linear system is composed of first degree functions, it is feasible to employ the Simplex algorithm, which works in a single or multi-objective way, finding optimal solutions for the linear system. This proposal encompasses items 3 and 4, the main question here is associated to the gain in the quality of the solution: given the currently representation model of the problem, to employ a multi-objective function will produce a qualitative gain proportional to the extra time spent?

[1] http://oaei.ontologymatching.org/

Finally, there are other questions more associated to the behavior of the final alignment method, where the efforts seeks to provide better stability to the algorithm in the case of successive executions. Nowadays it is possible to observe the need to work in an interdisciplinary way in computing, using knowledge from several areas to solve a single problem.

Working on the improvement of processes related to the linear system is fundamental to obtain better results in terms of time or quality, however, it is worth mentioning that the training instances of the model should reflect the behavior of the ontologies as best as possible, which means that if the ontologies have equivalences in the nomenclature of the entity, it is of great value that in the training instances there are correspondences that reflect those characteristics for a higher quality matching.

4. Final Considerations

Although there are several matchers and meta-matchers of ontologies in the literature, the application scenarios are not always the same, this work seeks to deal with meta-matching assigning candidate matchings to ontologies with few references and returning a first matching version to the engineers. Although it is in the initial stages, the preliminary results are interesting even with few adaptations of what was found in the literature. In addition to the expected contributions to the advancement of ontology meta-matchers, this work also contributes in the area of computational intelligence, since it presents an application scenario for the meta-heuristic of [Tilahun and Ong 2015].

The first step was given using the prey-predator meta-heuristic with continuous solution space, reaching f-measure rates of the matchings found in the 90% range. In addition to the proposed improvements, this work seeks to better understand the relationship between the problem and the forms that are used to model it, which representations have a greater descriptive capacity.

The study of the use of the prey-predator meta-heuristic to calibrate a set of matchers was submitted to Brazilian Database Symposium and it has been accepted. In the paper submitted, it is described the whole process of construction, movement and adaptation of the solutions, as well as the final matching methods after adjusting the weights.

References

Bock, J. and Hettenhausen, J. (2012). Discrete particle swarm optimisation for ontology alignment. *Information Sciences*, 192:152–173.

Farinelli, F. and Almeida, M. (2014). Interoperabilidade semântica em sistemas de informação de saúde por meio de ontologias formais e informais: um estudo da norma openehr. *XVII Encontro Nacional de Pesquisa em Ciência da Informação*, 17(1).

Otero-Cerdeira, L., Rodríguez-Martínez, F. J., and Gómez-Rodríguez, A. (2015). Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949–971.

Shi, Y. and Eberhart, R. (1998). A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 69–73. IEEE.

Sorensen, K., Sevaux, M., and Glover, F. (2017). A history of metaheuristics. *Handbook of Heuristics*.

Souza, J. F. (2012). *Uma abordagem heurística uni-objetivo para calibragem em meta-alinhadores de ontologias*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro.

Tilahun, S. L. and Ong, H. C. (2015). Prey-predator algorithm: A new metaheuristic algorithm for optimization problems. *International Journal of Information Technology & Decision Making*, 14(06):1331–1352.

Xue, X. and Tang, Z. (2017). An evolutionary algorithm based ontology matching system. *Journal of Information Hiding and Multimedia Signal Processing*.

Alinhamento de grandes Ontologias com recurso de banco de dados *NoSQL* e utilização de *workflow* científico

Luciana de Sá Silva Perciliano, Fernanda Araujo Baião Amorim (orientadora)

Departamento de Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Av. Pasteur, 458 – Rio de Janeiro– RJ – Brasil

{luciana.perciliano@uniriotec.br, fernanda.baiao@uniriotec.br}

Nível: Mestrado

Ingresso no Programa: 03/2017

Previsão de Defesa: 04/2019

Etapas concluídas: Definição do Problema, Proposta de Pesquisa apresentada em seminário na UNIRIO.

Publicações: Alinhamento de Ontologias com Suporte de um Sistema Gerenciador de *Workflows* Científicos – ERAD (IV Escola Regional de Alto Desempenho do Rio de Janeiro) – 05/2018.

Resumo. *Uma ontologia é um artefato que em um mesmo domínio pode ser representado de diferentes formas em sistemas distintos. O alinhamento de Ontologias é uma técnica que visa resolver o problema da heterogeneidade semântica entre esses sistemas. O alinhamento de grandes Ontologias (milhões ou bilhões de correspondências possíveis entre duas entidades de diferentes ontologias) é um desafio na área e demanda bastante tempo de execução e recursos computacionais. Esse desafio, pode ser verificado na iniciativa anual OAEI (Ontology Alignment Evaluation Initiative) que objetiva a comparação e avaliação dos sistemas de alinhamento. Dado que um banco de dados NoSQL é eficiente no armazenamento de grande volume de dados e que através de um Sistema de Gerência de Workflows Científicos (SGWfC) é possível a execução paralela e a execução em ambiente de nuvem é escalável, o objetivo da proposta de dissertação consiste em verificar se a utilização de desses recursos, resultam em menor tempo na execução e na escalabilidade em um processo de um sistema de alinhamento de Ontologias.*

1. Introdução

Uma ontologia, sob o viés computacional, é reconhecida como um artefato capaz de representar formalmente uma conceituação compartilhada [Gruber, 1993]. No entanto, existem diferentes formas de representar uma conceituação sobre um mesmo domínio, o que dificulta a troca de informações e entendimento entre diferentes sistemas, conhecido como problema da heterogeneidade semântica na ontologia. Uma das soluções para auxiliar esse problema é a aplicação de técnicas de alinhamento de Ontologias.

O alinhamento de Ontologias é um tópico que vem sendo pesquisado há algum tempo, com resultados positivos de bastante impacto e alguns desafios ainda em aberto. Dois desafios no alinhamento de Ontologias citados por [Shvaiko e Euzenat, 2013] são a eficiência e a escalabilidade. Os sistemas de alinhamento de Ontologias em sua maioria usam o armazenamento em memória e, com a existência cada vez mais frequente de ontologias de grande tamanho, há uma demanda bastante expressiva no consumo de memória, e o tempo requerido para o processamento das medidas de similaridades e técnicas correlatas, o que pode acarretar o elevado tempo de execução para seu término. A proposta de dissertação é apresentar uma estratégia para executar em paralelo as etapas de particionamento e de alinhamento de grandes ontologias baseada em Sistema de Gerência de *Workflows* Científicos (SGWfC) e Sistema Gerenciador de Banco de Dados (SGBD) *NoSQL* com o objetivo de aumentar o desempenho e a escalabilidade na execução de sistemas de alinhamento de Ontologias. Em particular, será aplicado o SGWfC SciCumulus [Silva et al. 2014] para execução paralela e o SGBD *NoSQL* Neo4J como recurso no processo de alinhamento de Ontologias.

Esse documento utiliza a seguinte disposição: Na seção 2 é realizado a Apresentação do Problema. A seção 3 descreve a Fundamentação Teórica. Nas seções 4 e 5 é descrito a Proposta da Solução e o Projeto de Avaliação da Solução. Na seção 6 os Trabalhos Relacionados. Na seção 7 é apresentada as Publicações. E na seção 8 a Conclusão.

2. Apresentação do Problema

A OAEI (*Ontology Alignment Evaluation Initiative*) é uma iniciativa anual e internacional que avalia e compara o desempenho dos sistemas de alinhamentos de Ontologias. A Figura 1 demonstra os *datasets* e os sistemas que participaram em 2017.

System	ALIN	AML	CroLOM	DiSMatch-ar	DiSMatch-sg	DiSMatch-tr	I-Match	KEPLER	Legato	LogMap	LogMap-Bio	LogMapLt	njuLink	ONTMAT	POMap	RADON	SANOM	Silk	WikiV2	XMap	YAM-BIO	Total=21
Confidence	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	-	✓	-	✓	✓	✓	✓	-	✓	-	16
anatomy	●	●	○	○	○	○	○	●	○	●	●	●	○	○	●	○	●	○	●	●	●	11
conference	●	●	○	○	○	○	○	●	○	●	○	●	○	●	●	○	●	○	●	●	○	10
largebio	○	●	○	○	○	○	○	◐	○	●	●	●	○	○	◐	○	◐	○	◐	●	●	10
phenotype	○	●	○	●	●	●	○	◐	○	●	●	◐	○	○	◐	○	○	○	○	◐	◐	11
multifarm	○	●	●	○	○	○	○	●	○	●	○	○	○	○	○	○	◐	○	●	◐	○	7
interactive	●	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	4
process model	○	●	○	○	○	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	3
instance	○	●	○	○	○	○	●	○	●	●	○	○	◐	○	○	○	○	○	○	○	○	5
hobbit ld	○	●	○	○	○	○	●	○	○	○	○	○	○	○	○	◐	○	◐	○	○	○	4
total	3	9	1	1	1	1	3	5	1	8	3	4	1	1	4	1	4	1	4	6	3	65

Figura 1. Resultado OAEI 2017 [Achichi et al. 2017]

As bolas pretas indicam que o sistema conseguiu executar com sucesso o alinhamento, as brancas significam que não foi executado e as metade preta e metade branca significam que não foi possível a conclusão da execução em um determinado *dataset* [Achichi et al. 2017]. De acordo com [Achichi et al. 2017]: "Não houve progresso significativo no que diz respeito à capacidade de sistemas de alinhamento para lidar com grandes ontologias e *datasets*, seja na correspondência de ontologia tradicional ou na correspondência de instâncias." e "Não houve melhora notável em relação aos tempos de execução do sistema". Os *datasets Disease* e *Phenotype Track* (*phenotype*) e *Large BioMed Track* (*largebio*) possuem ontologias de grande tamanho [Achichi et Al. 2017] que contêm respectivamente 138.143.706 e 1.224.924.624 correspondências máximas possíveis entre ontologias que compõe esses *datasets*. Alguns sistemas como: POMAP, SANOM, KEPLER e Wiki2, não terminaram a execução no tempo máximo atribuído pela iniciativa de quatro horas o alinhamento entre Ontologias do *dataset largebio* [Achichi et al. 2017]. Sendo assim, o problema do alinhamento de grandes Ontologias ainda é um desafio na área, conforme foi descrito em [Shvaiko e Euzenat, 2013].

3. Fundamentação Teórica

O alinhamento de Ontologias é o processo que, a partir de um par de ontologias, visa encontrar automaticamente o subconjunto de correspondências semânticas entre pares de entidades (classes, atributos, relacionamentos ou instâncias) das ontologias de entrada. Esse pode ser realizado de forma manual, semi-automática ou em tempo de execução. A Figura 2(a) ilustra um exemplo de alinhamento de Ontologias, no qual as classes estão colocadas dentro dos retângulos de cantos arredondados. As setas verticais exibem o relacionamento especialização-generalização, do mais específico "Literatura" para o mais genérico "Monografia". Os atributos estão colocados logo após as setas tracejadas. Nesse exemplo, as setas azuis representam as correspondências entre as duas ontologias (O1-Produto e O2-Monografia) [Shvaiko e Euzenat, 2013] [da Silva et. al. 2016]. Formalmente, uma correspondência é representada por um par de entidades e o tipo de relação existente entre elas pode ser: equivalência (=), disjunção (⊥) ou generalização (⊒). O alinhamento entre O1 e O2 é o conjunto das correspondências encontradas [Lopes, 2014]. No processo do alinhamento de Ontologias ilustrado na Figura 2(b), através das ontologias de entrada O1 e O2 é gerado o alinhamento A'. Esse pode ter passado por um outro processo de alinhamento A anteriormente.

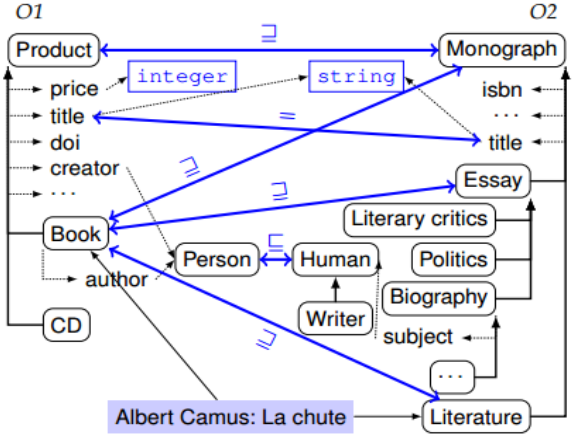


Figura 2(a). Alinhamento entre duas Ontologias [Shvaiko e Euzenat, 2013]

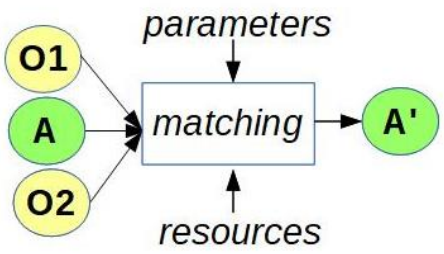


Figura 2(b). Processo de Alinhamento e Ontologias [Shvaiko e Euzenat, 2013]

Os parâmetros podem ser pesos e *thresholds*, e os recursos externos uma base léxica como a *Wordnet*, ontologias de referência ou relatórios de padrões e anti-padrões de alinhamento [Lopes, 2014], [da Silva et. al. 2016]. O sistema de alinhamento de Ontologias denominado ALIN [da Silva et al. 2016] busca aumentar a eficiência da participação do especialista (usuário conhecedor da ontologia que está sendo alinhada) e utiliza anti-padrões (combinação de correspondências que geram inconsistências) de alinhamento para melhorar a qualidade do alinhamento final. A utilização dessas características aumenta tanto a precisão como a cobertura do alinhamento obtido se comparado a abordagem que não as utiliza [da Silva et al. 2016]. A arquitetura do ALIN [da Silva et. al. 2016], utiliza os seguintes itens: APIs em Java (*Weka*, com rotinas estatísticas e de KDD), *Simmetrics* (métricas de similaridade baseadas em *strings*), WS4J (métricas linguísticas baseadas na *Wordnet*) e *Alignment* (possui rotinas para manipulação de ontologias escritas em OWL).

Nos bancos de dados *NoSQL* orientados a grafos, os dados são armazenados na forma de um grafo e servem para armazenamento de grandes volumes de dados e sua estrutura se assemelha ao esquema de uma ontologia. O banco de dados orientado a grafos Neo4J, possui uma estrutura requerida para armazenamento de ontologias [Pramanik, 2016]. No Neo4J, todas as entidades das ontologias (classes, propriedade de dados e propriedades de objetos) podem ser representadas pelos nós (vértices) e seus relacionamentos preservados através das arestas, assim como novos relacionamentos entre entidades de diferentes ontologias para representar correspondências no processo do alinhamento, com suas propriedades. A Figura 3, exemplifica relacionamentos entre entidades de duas diferentes ontologias do *dataset Conference* [Achichi et al. 2017]. Os nós representam as entidades, as arestas os relacionamentos, sendo que o relacionamento em destaque possui propriedades, por exemplo, NOME_DA_METRICA: “LIN” e VALOR_DA_METRICA: “0.16966131697611691” armazenadas no Neo4J.

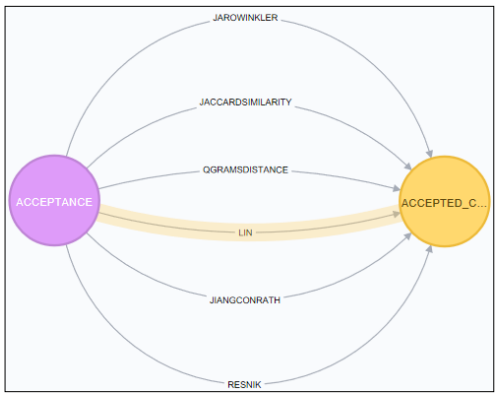


Figura 3. Exemplo de Ontologias no Neo4J

O processo do alinhamento de Ontologias no ALIN possui várias atividades, como na geração do conjunto de correspondências candidatas e na classificação e modificação do conjunto de correspondências candidatas que podem utilizar recursos, como o armazenamento em banco de dados. Esse processo se assemelha a um *workflow* científico que é uma forma abstrata de representar um experimento científico como uma sequência de atividades, sendo que o seu fluxo de dados pode sofrer diversas variações, incluindo a aplicação de parâmetros diferentes, conjuntos de dados de entrada distintos, algoritmos alternativos para a mesma tarefa, entre outros [de Oliveira et al. 2010]. Os Sistemas de Gerência de *Workflows* Científicos (SGWfC) executam e gerenciam os

fluxos de dados de um *workflow* científico [de Oliveira et al. 2010]. De acordo com [Silva & Mattoso, 2014]: "A gerência da dependência do fluxo de dados do *workflow* é um dos diferenciais dos SGWfC em relação a soluções que programam esse controle por *scripts* ou Hadoop, ou de forma independente (manual)". Diferentemente do conceito de MapReduce em que é necessário para os cientistas programarem a execução de experimento científico [de Oliveira et al. 2010], o SciCumulus é um *middleware* que promove uma execução de um *workflow* em um ambiente de nuvem, sendo possível controlar e monitorar as execuções das atividades desse, de forma a isolar o usuário da complexidade de um ambiente de nuvem e da distribuição dos dados. Além disso, um benefício no uso do SciCumulus é o banco de proveniência que permite aos usuários a consulta dos dados históricos, análise e monitoração do fluxo de dados em todo seu ciclo de vida inclusive em tempo de execução [de Oliveira et al. 2010] [Silva & Mattoso, 2014]. Diversos SGWfC existentes na literatura têm suporte para acesso intensivo a dados aplicando técnicas de processamento de alto desempenho, incluindo suporte a paralelismo e distribuição [Liu et al. 2015], por exemplo, Pegasus e Swift, nesses SGWfC os dados de proveniência são disponibilizados apenas no final da execução do *workflow* [Silva & Mattoso, 2014].

4. Proposta de Solução

A proposta de solução é utilizar um SGWfC em particular o SciCumulus para executar, monitorar e controlar a execução de um sistema de alinhamento de Ontologias em um ambiente de nuvem de forma paralela, pelos motivos descritos na seção 3. E a utilização do recurso de armazenamento de dados de ontologias no Neo4J (conforme modelagem descrita na seção 3) no processo de alinhamento de Ontologias. A contribuição principal desta proposta é uma estratégia de alinhamento de grandes ontologias que executa em paralelo as etapas de particionamento e de *matching* das ontologias. A figura 4 ilustra a arquitetura da solução. Dado 2 ontologias O1 e O2, elas passam por um processo de "*partition of ontologies*" em que serão particionadas, ou seja, ao invés de todos os pares das entidades de O1 serem alinhados com os de O2, eles serão divididos em vários blocos. Depois dessa divisão, por exemplo, O1' e O2', passa pelo processo de "*matching*" que recebe "*parameters*" e em que "*resources*" podem ser consultados como um banco *NoSQL* sempre que necessário, essa etapa gera um alinhamento parcial A'. Isso ocorre até que todos os pares de subontologias de O1 e O2 gerem o alinhamento parcial A'. Quando todos os alinhamentos parciais A' estiverem finalizados, esses passam pelo "*combine matching*" em que todos os alinhamentos A' serão combinados, gerando um único alinhamento final das correspondências entre O1 e O2 denominado AF.

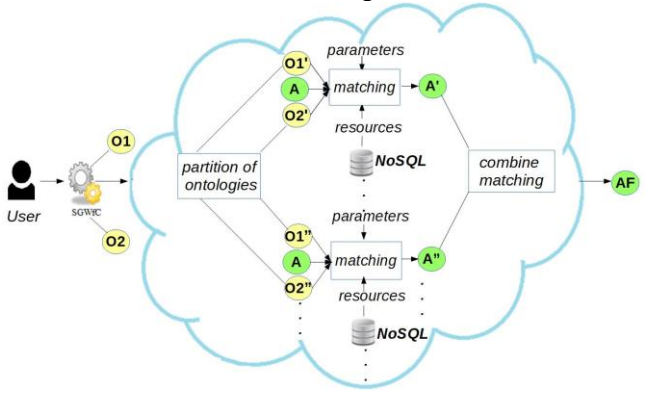


Figura 4. Arquitetura da Solução

5. Projeto de Avaliação da Solução

Utilizar o SGWfC SciCumulus [Silva et al. 2014] para gerenciar a execução paralela em um ambiente de nuvem do sistema de alinhamento de Ontologias ALIN [da Silva et al. 2016] com os *datasets phenotype* e *largebio* disponíveis na página do OAEI 2017[1], com a utilização de recurso no processo de alinhamento o banco de dados orientado a grafos Neo4J. A precisão, cobertura e medida-F são formas de avaliar a qualidade do alinhamento de Ontologias que foi gerado [da Silva et al. 2016]. Os resultados serão avaliados com relação ao tempo de execução e também com relação à precisão, cobertura e medida-F (de forma a garantir que a execução em paralelo não impactou a qualidade dos resultados encontrados) comparando-os com os resultados do OAEI 2017.

6. Trabalhos Relacionados

Existem alguns trabalhos que utilizam a execução paralela em nuvem para diminuir o tempo de execução do alinhamento de Ontologias, como em [Araújo et al. 2015] que utiliza o particionamento das ontologias em subontologias e a abordagem MapReduce no processo do alinhamento das Ontologias, um dos seus trabalhos futuro é investigar como grandes ontologias podem ser particionadas em paralelo com algoritmo de particionamento PAP (*Partition, Anchor, Partition*). Em [Araújo et al. 2016] é proposto uma técnica de *load balancing* e a abordagem MapReduce para o alinhamento de grandes ontologias e possui como um dos seus trabalhos futuro a pesquisa de como particionar em paralelo grandes ontologias.

7. Publicações

As tabelas 1 e 2 apresentam respectivamente as principais publicações relacionadas e o plano de publicação.

Tabela 1. Principais publicações relacionadas

Publicação
Achichi, M., Cheatham, M., Dragisic, Z., Euzenat, J., Faria, D., Ferrara, A., ... & Zamazal, O. (2017*). Results of the Ontology Alignment Evaluation Initiative 2017*. In Proceedings of the 12th International Workshop on Ontology Matching-Volume 2032 (pp. 61-113)
Araújo, T. B., Pires, C. E., da Nobrega, T. P., & Nascimento, D. C. (2015). *A parallel approach for matching large-scale ontologies*. Journal of Information and Data Management, 6(1), 18.
da Silva, J., Baião, F. A., & Revoredo, K. (2016). Alinhamento Interativo de Ontologias usando Anti-Padrões de Alinhamento: Um primeiro Experimento. XII Brazylian Symposiumon Information Systems, Florianopolis, SC, p. 208-215.
de Oliveira, D., Ogasawara, E., Baião, F., & Mattoso, M.(2010). Scicumulus: *A light-weight cloud middleware to explore many task computing paradigm in scientific workflows*. IEEE 3rd International Conference on(pp. 378-385).
Shvaiko, P., & Euzenat, J. 2013. *Ontology matching: state of the art and future challenges*.IEEE Transactions on Knowledge and Data Engineering, 25(1), p. 158-176.

Tabela 2. Plano de publicação

Evento/ Periódico	Data prevista de submissão	Conteúdo do trabalho	Situação
SBBD-2019	05/2019	Resultados alcançados comparando com resultados do OAEI 2018	não-submetida
Semantic Web journal	12/2018	Resultados do alinhamento de grandes Ontologias com SGWfC e banco de dados *NoSQL*	não-submetida

[1] *http://oaei.ontologymatching.org/2017/*

8. Conclusão

O alinhamento de grandes Ontologias ainda é um desafio, conforme [Achichi et al. 2017]. A proposta de dissertação propõe uma estratégia que executa em paralelo as etapas de particionamento e de alinhamento de grandes Ontologias e utiliza o SGWfC SciCumulus para que esse ocorra em um ambiente de nuvem de forma paralela e também o banco de dados *NoSQL* Neo4J como recurso no processo de alinhamento, para comprovação da hipótese a ser testada. Presume-se que as características dessas tecnologias podem contribuir para a diminuição do tempo de execução e na escalabilidade de um sistema de alinhamento de grandes Ontologias.

Referências

Achichi, M., Cheatham, M., Dragisic, Z., Euzenat, J., Faria, D., Ferrara, A., ... & Zamazal, O. (2017). *Results of the Ontology Alignment Evaluation Initiative 2017*. In Proceedings of the 12th International Workshop on Ontology Matching-Volume 2032 (pp. 61-113)

Araújo, T. B., Pires, C. E., da Nobrega, T. P., & Nascimento, D. C. (2015). *A parallel approach for matching large-scale ontologies*. Journal of Information and Data Management, 6(1), 18.

Araújo, T. B., Pires, C. E. S., da Nóbrega, T. P., & Nascimento, D. C. (2016). *A fine-grained load balancing technique for improving partition-parallel-based ontology matching approaches*. Knowledge-Based Systems, 111, 17-26.

da Silva, J., Baião, F. A., & Revoredo, K. (2016). Alinhamento Interativo de Ontologias usando Anti-Padrões de Alinhamento: Um primeiro Experimento. XII Brazylian Symposium on Information Systems, Florianopolis, SC, p. 208-215.

de Oliveira, D., Ogasawara, E., Baião, F., & Mattoso, M., 2010. Scicumulus: *A light-weight cloud middleware to explore many task computing paradigm in scientific workflows*. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on (pp. 378-385). IEEE.

Gruber, T. R. (1993). *A translation approach to portable ontology specifications*. Knowledge acquisition, 5(2), 199-220

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

Lopes, V., 2014. Alinhamento Interativo de Ontologias: Uma Abordagem Baseada em *Query-by-Committee* (Doctoral dissertation, Msc Dissertation, Rio de Janeiro, Rj, Brazil: Universidade do Estado do Rio de Janeiro (UNIRIO)).

Pramanik, Gopal. (2016). *Ontology and Neo4J Graph Database*. Scientific Voyage, Volume -2, Issue - 2, Date Of Publication - May, 2016.

Shvaiko, P., & Euzenat, J. (2013*). Ontology matching: state of the art and future challenges*. IEEE Transactions on Knowledge and Data Engineering, 25(1), p. 158-176

Silva, V., Oliveira, D., & Mattoso, M. (2014). SciCumulus 2.0: Um Sistema de Gerência de *Workflows* Científicos para Nuvens Orientado a Fluxo de Dados. Sessão de Demos do XXIX Simpósio Brasileiro de Banco de Dados.

Effective method for detecting drunk texting

Marcos A. Grzeça[1], **Karin Becker**[1], **Renata. Galante**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{magrzeca,karin.becker,galante}@inf.ufrgs.br

Nível: Mestrado
Data de ingresso no mestrado: Março de 2017
Data prevista para conclusão do mestrado: Março de 2019
Etapas concluídas: revisão da literatura; identificação do baseline; definição e implementação do modelo de enriquecimento semântico.
Etapas futuras: integração entre word embeddings e enriquecimento semântico; experimentação; avaliação.

Abstract. *Data from social networks is used by researchers to analyze social problems due to the ease in obtaining up-to-date data. A current social problem is excessive alcohol consumption. Social networks are key factors to understand the reasons that lead to excessive consumption, since the use of social networks and alcohol consumption are behaviors linked to young people. Classifying drunk texting from social networking data is complex because texts are short, sparse and written with diverse vocabulary. The aim of this work is to provide an efficient method to identify drunk texting through a framework that combines semantic enrichment and word embeddings. Developing a framework combining these techniques is important to analysis of short texts, because semantic enrichment provides features that add context to the texts, but adding features to word embeddings without interfering with context is still an open problem.*

Resumo. *Dados de redes sociais são utilizados por pesquisadores para analisar problemas sociais devido a facilidade em obter dados atualizados. Um problema social atual é o consumo excessivo de álcool. Redes sociais são fatores chaves para entender os motivos que levam ao consumo excessivo, visto que a utilização de redes sociais e o consumo de álcool são comportamentos vinculados a jovens. Classificar drunk texting a partir de dados de redes sociais é complexo, pois os textos são curtos, esparsos e escritos com um vocabulário diversificado. O objetivo deste trabalho é fornecer um método eficaz capaz de identificar drunk texting através de um framework que combine enriquecimento semântico e word embeddings. Desenvolver um framework combinando essas técnicas é importante para analisar textos curtos, pois o enriquecimento semântico fornece features que adicionam contexto aos textos, mas adicionar as features ao word embeddings sem interferir no contexto ainda é um problema em aberto.*

1. Introduction

Researchers have investigated the use of social networks and big data to discover information regarding topics related to health, social problems, and events [Kershaw et al. 2014]. Social networks are often used as basis of these studies due to the data being generated in large volumes, up-to-date, easy to obtain [Culotta 2013].

Excessive alcohol consumption is a major problem that affects our society. Alcohol caused around 3.3 million deaths worldwide in 2012 and is responsible for 15% of deaths resulting from traffic accidents [Organization and of Substance Abuse Unit 2014]. Young people are more likely to end up in fatal traffic accidents linked to alcohol consumption [de Informações sobre Saúde e Álcool 2014]. It is important to understand and monitor the factors that cause high alcohol consumption to assist in public health policies.

Social networks are key elements to this purpose, since alcohol consumption and the use of social networks are behaviors related to youngsters. Young people who use social networks are more likely to use tobacco, alcohol, and marijuana [Johnson et al. 2011] — and repeated exposure to posts involving drugs on social networks may encourage others, as this behavior may then be perceived as normal [West et al. 2012]. Texting while under the influence of alcohol is popularly regarded as drunk texting[1]. In Twitter, tweets written under the influence of alcohol are 'drunk tweets', and the opposite are 'sober tweets' [Joshi et al. 2015].

Official data on alcohol consumption and data extracted from drunk tweets are strongly correlated [Culotta 2013, Kershaw et al. 2014], confirming the accuracy of inferences extracted from social networks. Moreover, about 500 million tweets are posted daily, and 37% of registered users are between 18 and 29 years old. Thus, Twitter can be a valuable source of information to this end [West et al. 2012] because it provides messages, mostly posted by young people, which may be related to alcohol consumption and can be obtained in real time.

The automatic identification of drunk texting allows detailed studies on alcohol consumption based on a large volume of data and can warn family or friends about some danger (e.g. driver is driving while drunk). Such studies can provide public officials with information related to the identification of factors that cause excessive consumption of alcohol, helping in its prevention and control. Techniques suitable for drunk texting classification can be adapted to identify other types of drug abuse.

The classification of drunk texting is a complex task, because texts are written in natural language and different vocabulary with a great morphological variation. Another issue is the scarcity of categorized databases that can be manipulated for this goal. Related work has developed drunk tweets classifiers based on features extracted using traditional natural language processing (NLP) techniques (e.g. n-grams, stemming), sentiment analysis, and the morphology of the sentence to identify alcohol consumption [Jauch et al. 2013, Aphinyanaphongs et al. 2014, Joshi et al. 2015, Hossain et al. 2016]. None of these works have attempted to improve the categorization text by providing contextual meaning through semantic enrichment.

The objective of this work is to develop a method capable of identifying drunk texting by combining two very active research areas: semantic enrichment ([Romero and Becker 2017]) and word embeddings ([Li et al. 2016]). The research ques-

[1] https://www.urbandictionary.com/define.php?term=drunk+texting

tions that guide our work are: *i*) How to semantically enrich tweets and to identify the most important entities; *ii*) What is the best way to use word embeddings in this context? and *iii*) How can word embeddings and semantic enrichment complement each other?

The combined use of word embeddings and semantic enrichment is important for analyzing short and noisy pieces of text, especially when there is not a large number of records available. Under these conditions, the exclusive use of word embeddings may result in the learning misleading or irrelevant patterns found in the training data [Chollet and Allaire 2018]. On the other hand, the single use of semantic enrichment is not enough to learn all relationships between the tokens present in the texts.

The remainder of this paper is structured as follows: Section 2 presents related works linked to the area. Section 3 introduces the proposed method. Section 4 presents the preliminary results obtained. Section 5 addresses future works.

2. Related work

The detection of drunk texting is inserted in the area of paralinguistics. This area began to receive increased attention from researchers in 2011 with the challenge 'The INTERSPEECH 2011 Speaker State Challenge' ([Schuller et al. 2011]) where the goal was to classify the intoxication level through speech.

Since then, drunk texting classification has been addressed by works such as [Jauch et al. 2013], [Aphinyanaphongs et al. 2014], [Joshi et al. 2015] and [Hossain et al. 2016]. These works use traditional feature extraction techniques for text classification, such as classical pre-processing (e.g. tokenization, removal of stopwords, normalization of mentions/URLs), data cleansing, extraction of n-grams, stylistic features (e.g. number of discourse connectors, number of words, number of capital letters in the tweet), and sentiment analysis. They also experimented with different classification algorithms, such as Random Forest, Support Vector Machine (SVM), Generalized Linear Model (GLM), and Naive Bayes.

The existing papers in this area did not use semantic enrichment nor word embeddings strategies. The developed method allows extracting components that indicate the consumption of alcohol, enabling more detailed studies that help to understand reasons for alcohol abuse.

2.1. Semantic enrichment

To achieve good results in text classification problems, it is necessary that the text be long enough to provide features for the classifier to learn patterns in the data [Li et al. 2016]. To deal with short and sparse texts, such as tweets, related work uses semantic enrichment, because it improves the identification of representative terms or entities related to a tweet [Romero and Becker 2017].

Semantic enrichment uses external knowledge bases (DBPedia, Wikipedia, YAGO, etc...) to add context to short texts and extracts named entities (NER), key words, concepts, and categories present in the text. Entities can generalize texts written with different terms, but with the same semantic meaning.

2.2. Word embeddings

The classification of short texts also suffers from the problem of 'word mismatch', due to its limited context [Li et al. 2016]. To overcome word mismatch, word embeddings can be

used. Word embeddings are dense vector representation of words whose value can be used to determine the similarity between words. However, as pointed out by [Li et al. 2016], 'to get high quality embedding vectors, a large amount of training data is necessary'.

While word embeddings are useful to determine the similarity of words, semantic enrichment contributes by adding context to the text. There is an opportunity to integrate them to analyze short texts because the addition of semantic enrichment allows word embeddings to learn semantic similarity of texts, helping in the classification of short texts written with different words, but with similar meanings. The main challenge in using semantic enrichment and word embeddings is how to add semantic information without interfering in the context of texts, damaging the vector representation of word embeddings.

3. Proposed Framework

The goal is to develop a method that can identify drunk texting in short and noisy pieces of text, such as tweets, by integrating semantic enrichment and word embeddings. Semantic enrichment can provide context to texts, while word embeddings learning the similarity between the words in text, optimizing the classification of short texts.

Initially, semantic enrichment (Section 3.1) and word embeddings (Section 3.2) have been used alone for text classification. We are working to integrate semantic enrichment and word embeddings.

3.1. Semantic enrichment

In this section, we describe the proposed framework for improving the classification of drunk texting in tweets using semantic enrichment, illustrated in Figure 1. This framework is based on the strategies adopted by [Romero and Becker 2017] for event classification.

The framework is composed of seven steps: *i*) pre-processing, which includes emoticon polarity and posting time identification; *ii*) error handling to deal with typing errors that are a possible side effect of drunkenness; *iii*) extraction of conceptual features using Natural Language Understanding (NLU) to provide meaning to terms; *iv*) use of linked data (from DBPedia) to extract semantic aspects that can be used to generalize the conceptual features; *v*) pruning to select only the discriminant features, since the NLU and linked data stages result in a large number of features, which may degrade the classification performance; *vi*) textual features (1-grams, bi-grams, and hashtags), flags indicating

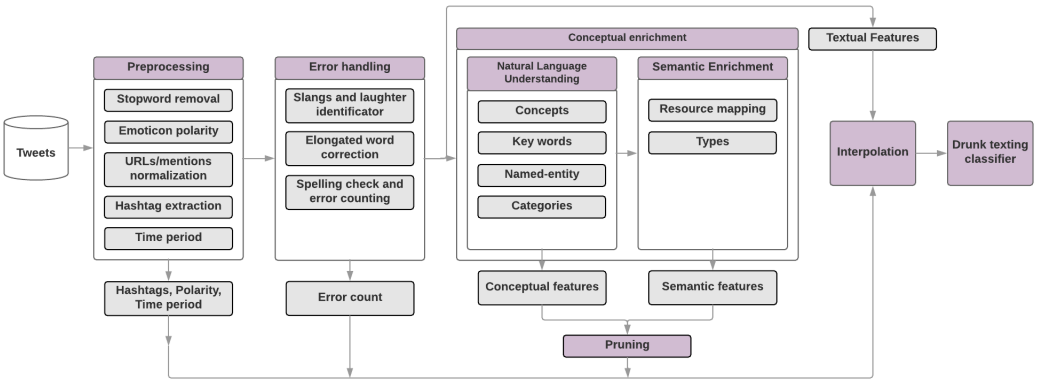


Figure 1. Framework Overview

the presence of errors, posting time and conceptual/semantic features are combined; *vii*) tweets are classified into either: 'drunk texting' or 'sober texting'.

3.2. Word embeddings

We propose to learn the word embeddings from a tweet text using the *word2vec* algorithm. In the current stage of the research, we use these embeddings as one of the layers of a recurrent neural network Long Short-Term Memory (LSTM) to classify drunk tweets. As the next step, we will investigate how to integrate semantic enrichment and word embeddings, and experiment with other LSTM architectures.

4. Experiments and preliminary results

The goal of our experiments is to verify the effectiveness of the proposed framework towards identifying drunk texting. We assess our contribution by comparing metrics of information retrieval (IR) with the results obtained by [Hossain et al. 2016].

4.1. Dataset

The dataset is based on [Hossain et al. 2016]. This article was chosen as baseline because it made available its annotated dataset for tests. We collected 3996 tweets. For each tweet in the dataset, [Hossain et al. 2016] they hired Amazon Mechanical Turk[2] and asked three questions: *Q1*) Tweet mentions the activity (drinking alcohol); *Q2*) Tweet is about the tweeter doing the activity (user drinking alcohol) and *Q3*) Tweet is about user doing the activity when tweeting (user tweeting while drunk).

4.2. Goals

Our goal is to develop one classifier for each label available [Hossain et al. 2016]. With *Q1* we want to evaluate the performance of the framework for classifying tweets that mention alcohol consumption. However, *Q1* may also contain tweets related to news or ads linked to alcohol consumption. Therefore, with *Q2* we want to evaluate the effectiveness of the framework for identifying tweets in which the user is consuming alcohol. Finally, with *Q3* we can classify tweets where the user is tweeting while drunk. We consider *Q2* and *Q3* classifiers more relevant with regard to our goal of drunk texting classification, since *Q1* is annotated for the mere mention to alcohol (including ads).

4.3. Experiments

Experiments with semantic enrichment and word embeddings have been performed exclusive of each other. The results of the experiments with semantic enrichment were submitted to Web Intelligence 2018 for evaluation. On the other hand, experiments with word embeddings are in the initial stage and did not present good results, possibly due to the small dataset.

4.3.1. Experiment 1: Semantic enrichment

We run our experiments using the classification algorithm SVM Poly with cross-validation of 5 folds, reserving 80% of the data for training and 20% for tests. We run each experiment 10 times, and performed a statistical paired t test using $\alpha = 0.05$. Table 1

[2]http://www.mturk.com

Table 1. Results with semantic enrichment

	Recall	Precision	F1	+ pp in Recall	+ pp in Precision	+ pp in F1
Q1	87.517	92.151	89.834	-0.157	3.104	1.474
Q2	96.715	81.398	89.057	7.372	-0.264	3.553
Q3	95.182	80.892	88.037	13.79	4.701	9.249

Table 2. Results with Word Embeddings and LSTM (Q3)

Method	F1	Precision	Recall
Baseline	81.392	81.392	81.392
SVM + Semantic Enrichment	88.037	80.892	95.182
Neural Network LSTM	73.737	75.376	72.699
Neural Network LSTM + Semantic Enrichment	76.125	74.483	78.156

summarizes the results obtained with our semantic enrichment framework. The gains in percentage points (pp) with regard to the baseline are highlighted in the columns with a '+' symbol. The most significant ones are observed for recall in actual drunk texting situations (*Q2* and *Q3*, with 7.372 and 13.79 pp, respectively). The results of *Q2* are statistically significant for all metrics. With regard to *Q3*, the results are statistically significant only for recall and F1-measure. The importance of the proposed features was confirmed, as semantic/conceptual features account for 63-73% of the 30 most relevant features for classification *Q2* and *Q3*.

4.3.2. Experiment 2: Word embeddings

Two experiments have been performed to develop a Q3 classifier. The first one considered only the embeddings of the tweets, while the second experiment used the embeddings of the tweets concatenated the embeddings of the semantic enrichment. We learned the embeddings from tweet text, and generated vectors of 32 dimensions. In our experiments the neural network has two intermediate layers with 16 hidden layers each and uses the activation function *reLU*. The results are shown in Table 2, and were inferior compared to the baseline and semantic enrichment only. We hypothesize that these poor results may be due to the size of the dataset from which the embeddings were extracted. We are currently working on the expansion of this dataset in order to perform new experiments.

5. Next Steps

The semantic enrichment framework enabled us to outperform the baseline, and we are currently developing more experiments with word embeddings to verify whether the poor results can be improved with a larger dataset and alternative neural network architectures.

To address the last research question, we are investigating how to combine word embeddings with semantic enrichment. To this end, we are experimenting different architectures of LSTM and CNN neural networks. Such neural networks have layers of embeddings for the text of tweets and auxiliary inputs for semantic enrichment, so that the embedding vector is concatenated to the auxiliary vector. Through the experiments, it will be possible to verify if these neural networks are adequate to combine with word embeddings and semantic enrichment. The use of these strategies together can provide better results in the analysis of short texts in general, not just the identification of drunk texting.

References

Aphinyanaphongs, Y., Ray, B., Statnikov, A., and Krebs, P. (2014). Text classification for automatic detection of alcohol use-related tweets: A feasibility study. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*. IEEE.

Chollet, F. and Allaire, J. J. (2018). *Deep Learning with R*. Manning Publications Company.

Culotta, A. (2013). Lightweight methods to estimate influenza rates and alcohol sales volume from twitter messages. *Language resources and evaluation*, 47(1):217–238.

de Informações sobre Saúde e Álcool, C. (2014). Álcool e trânsito. http://www.cisa.org.br/artigo/4692/alcool-transito.php.

Hossain, N., Hu, T., Feizi, R., White, A. M., Luo, J., and Kautz, H. A. (2016). Precise localization of homes and activities: Detecting drinking-while-tweeting patterns in communities. In *ICWSM*, pages 587–590.

Jauch, A., Jaehne, P., and Suendermann, D. (2013). Using text classification to detect alcohol intoxication in speech. In *Proceedings of the 7th Workshop on Emotion and Computing at the 36th German Conference on Artificial Intelligence*.

Johnson, T., Shapiro, R., and Tourangeau, R. (2011). National survey of american attitudes on substance abuse xvi: Teens and parents. *The National Center on Addiction and Substance Abuse*, 2011.

Joshi, A., Mishra, A., AR, B., Bhattacharyya, P., and Carman, M. J. (2015). A computational approach to automatic prediction of drunk-texting. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics.

Kershaw, D., Rowe, M., and Stacey, P. (2014). Towards tracking and analysing regional alcohol consumption patterns in the uk through the use of social media. In *Proceedings of the 2014 ACM conference on Web science*, pages 220–228. ACM.

Li, Q., Shah, S., Liu, X., Nourbakhsh, A., and Fang, R. (2016). Tweetsift: Tweet topic classification based on entity knowledge base and topic enhanced word embedding. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2429–2432. ACM.

Organization, W. H. and of Substance Abuse Unit, W. H. O. M. (2014). *Global status report on alcohol and health, 2014*. World Health Organization.

Romero, S. and Becker, K. (2017). Improving the classification of events in tweets using semantic enrichment. In *Proceedings of the International Conference on Web Intelligence*, pages 581–588. ACM.

Schuller, B., Steidl, S., Batliner, A., Schiel, F., and Krajewski, J. (2011). The interspeech 2011 speaker state challenge. In *Twelfth Annual Conference of the International Speech Communication Association*.

West, J. H., Hall, P. C., Hanson, C. L., Prier, K., Giraud-Carrier, C., Neeley, E. S., and Barnes, M. D. (2012). Temporal variability of problem drinking on twitter. *Open Journal of Preventive Medicine*, 2(01):43.

Publicação de Dados Abertos Conectados Sobre os Transplantes Realizados no IMIP

Aluna: Rayelle Ingrid Vera Cruz Silva Muniz

E-mail: rivcs@cin.ufpe.br

Orientadora: Bernadette Farias Lóscio

E-mail: bfl@cin.ufpe.br

Universidade Federal de Pernambuco - UFPE

Programa de Pós-Graduação em Ciência da Computação – Centro de Informática

Nível: Mestrado

Mês e ano de ingresso: março/2017

Mês e ano previstos para defesa: março/2019

Etapas Concluídas: Créditos em disciplinas, Definição do Problema, Especificação e Referencial Bibliográfico Inicial

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

Abstract. *Universities, governments, companies, startups and many other organizations are increasingly using the Web as the primary means of sharing and generating content. One of the areas that have shown interest in publishing data in open format is the Health's area. With the dissemination of the Semantic Web and the Linked Data principles we can find several clinical studies, relevant analyzes of hospital data and various other valuable information being published in open format. Given this scenario, this work aims the creation of a linked open dataset regarding the transplants performed in IMIP.*

Resumo. *Universidades, governos, corporações, startups e diversas outras organizações estão cada vez mais utilizando a Web como principal meio de compartilhamento e geração de conteúdo. Com isso, uma das áreas que têm demonstrado interesse na publicação de dados abertos é a área da Saúde. Com a disseminação da Web Semântica e dos princípios de Linked Data podemos encontrar diversos estudos clínicos, análises relevantes de dados hospitalares e diversas outras informações de valor sendo publicadas em formato aberto. Diante desse cenário, este trabalho tem por objetivo a criação de um conjunto de dados abertos e conectado a respeito dos transplantes realizados no IMIP.*

Palavras-Chave: *Dados Abertos, Dados Abertos Conectados, Linked Data, Publicação e Consumo de Dados, Dados de transplantes, Dados na Web.*

1. Introdução e Motivação

A Web é hoje o principal meio de compartilhamento de informações. O crescimento do número de fontes de dados na Web tem aumentado o interesse de organizações, instituições de ensino e governos a publicar seus dados em formato aberto, além de motivar o desenvolvimento de aplicações e ferramentas que consomem o conteúdo disponibilizado por estas fontes.

O interesse na utilização da Web como plataforma para publicação de dados não é algo novo [Berners-Lee *et al.* 1999, Abiteboul *et al.* 2000]. Porém, nos últimos anos este interesse tem aumentado devido à flexibilidade que a Web provê para a publicação e consumo de dados. Em sites como o DataHub[1] é possível encontrar diversas fontes de dados em formato aberto, o que confirma o grande potencial que a Web tem para a publicação e o consumo de dados e o constante interesse dos provedores de dados em utilizá-la.

A publicação de dados em formato aberto tem gerado benefícios em diversas áreas, como a transparência dos órgãos governamentais, que provê melhor acesso aos dados, participação e colaboração da sociedade em seus governos [W3C Escritório Brasil 2011]. A abertura dos dados também pode contribuir para o avanço da ciência, bem como a geração de novos empregos; pode aumentar, também, a qualidade nos serviços prestados por diversas organizações, além de muitos outros benefícios que podem ser encontrados em [Pires 2015].

Além do setor governamental, uma das áreas que têm demonstrado interesse na publicação de dados abertos é a área da Saúde. Muitas instituições de saúde e/ou provedores de dados independentes, têm criado *datasets* que possibilitam a geração de análises relevantes. Como exemplo, podemos encontrar no DataHub o *dataset* "*Pharmaceutical Drug Spending by Countries*[2]" que possui indicadores sobre o total de gastos com drogas farmacêuticas para saúde por país. Ainda podemos encontrar o *dataset* "*Breast Cancer*[3]" com dados sobre as ocorrências de câncer de mama. Por fim, podemos mencionar o *dataset* "*Cervical Cancer*[4]" com dados sobre as ocorrências de câncer cervical nas mulheres.

Porém, infelizmente em alguns casos os dados a serem publicados em formato aberto encontram-se em sistemas proprietários ou até mesmo em registros físicos. Nesses casos, o processo de publicação requer algumas etapas adicionais que podem envolver desde a digitalização dos dados até a modelagem dos conjuntos de dados a serem publicados.

Nesse contexto, este trabalho tem como objetivo a criação de um conjunto de dados sobre os transplantes realizados no Instituto de Medicina Prof. Fernando Figueira (IMIP)[5]. Atualmente, o cadastro dos pacientes candidatos a um transplante, bem como

[1] https://datahub.io

[2] https://datahub.io/core/pharmaceutical-drug-spending

[3] https://datahub.io/machine-learning/breast-cancer

[4] https://datahub.io/machine-learning/cervical-cancer

[5] http://www1.imip.org.br/imip/home/index.html

os dados dos pacientes transplantados são realizados manualmente. O setor de transplantes do IMIP conta com os dados de mais de 1600 pacientes transplantados e mais de 500 pacientes aguardando na fila por um transplante. Considerando que os primeiros registros foram feitos há mais de 15 anos atrás, o volume de dados disponível é grande e pode ser bastante útil para a realização de análises, como a identificação do perfil dos pacientes transplantados com maior taxa de sobrevivência.

É importante observar que o conjunto de dados, resultado deste trabalho, será criado de acordo com as boas práticas para dados na Web (*Data on the Web Best Practices).* [Lóscio *et al.* 2017], propõe um conjunto de boas práticas com o intuito de produzir conjuntos de dados de qualidade facilitando uma melhor comunicação entre provedores e consumidores de dados. Além disso, será criado um vocabulário para a descrição dos metadados estruturais. O uso desse vocabulário também facilitará a publicação dos dados de acordo com os princípios de *Linked Data*.

O restante deste artigo está organizado como se segue: a Seção 2 introduz alguns conceitos; a Seção 3 descreve a solução proposta; a Seção 4 apresenta a metodologia utilizada para a realização deste trabalho; a Seção 5 discute alguns trabalhos relacionados, e a Seção 6 traz algumas considerações, indicando os próximos passos para sua conclusão.

2. Fundamentação Teórica

O conceito de Dados Abertos aplica-se a todo dado publicado na Web disponível para que qualquer usuário possa utilizar, reutilizar e redistribuir esse dado sem qualquer restrição de patentes, propriedade intelectual ou outro mecanismo de controle, estando sujeito, no máximo, a atribuição de autoria.

Linked Data (Dados Conectados), por outro lado, é um conjunto de princípios para publicação de dados estruturados. Um dos principais objetivos do *Linked Data* é prover uma Web onde os dados possam estar diretamente ligados com outros dados por meio de *links* RDF, possibilitando a navegação entre diferentes conjuntos de dados e permitindo a realização de inferências.

A integração desses dois conceitos resulta nos dados abertos conectados ou Linked Open Data [Isotani and Bittencourt 2015], tornando mais fácil a manipulação e reutilização dos dados, agregando valor e possibilitando a descoberta de novos dados vinculados. É importante ressaltar que dados conectados não necessariamente precisam ser abertos.

O processo de publicação e consumo de dados na Web envolve várias fases que vão desde a preparação dos conjuntos de dados a serem publicados até o *feedback* sobre os dados utilizados e o refinamento dos dados gerados. Esse conjunto de fases que compõe o processo de publicação e consumo dos dados é chamado de Ciclo de Vida dos Dados na Web.

As fases do ciclo de vida representado na Figura abaixo são brevemente descritas a seguir [Lóscio *et al.* 2015]:

Ciclo de Vida dos Dados na Web

1. **Preparação:** A primeira fase começa desde o momento em que há a intenção de se publicar os dados e se estende até a seleção dos dados que serão publicados.
2. **Criação:** Esta etapa é a de extração dos dados de fontes de dados já existentes até a sua transformação para o formato adequado para publicação na Web.
3. **Avaliação:** Esta etapa requer a avaliação de especialistas da área em que se quer publicar os dados, a fim de que eles possam certificar a qualidade dos mesmos.
4. **Publicação:** Após a avaliação dos especialistas, os dados serão disponibilizados de forma pública na Web, sendo importante a garantia ao usuário de que os dados serão atualizados de acordo com uma frequência pré-determinada, a qual deverá ser disponibilizada juntamente com os dados.
5. **Consumo:** Nesta fase os dados estão disponíveis para serem utilizados para a criação de visualizações, como gráficos, bem como aplicações que permitam a realização de análises sobre os dados.
6. ***Feedback:*** Uma das fases e maior importância, pois é a partir do *feedback* dos usuários que é possível identificar melhorias e realizar correções nos dados previamente publicados.
7. **Refinamento:** Esta fase compreende todas as atividades relacionadas a adições ou atualizações nos dados que já foram publicados. É de suma importância garantir a manutenção e correção dos dados, de acordo com os *feedbacks* recebidos pelos consumidores, a fim de oferecer maior segurança para os consumidores dos dados.

3. Solução Proposta

O objetivo geral deste trabalho é a publicação de um conjunto de dados abertos conectados sobre os transplantes realizados no IMIP, a ser criado de acordo com as melhores práticas propostas em [Lóscio *et al.* 2017], bem como a criação de um vocabulário para a descrição dos metadados estruturais.

Atualmente, as informações sobre os pacientes transplantados e os que estão na fila de espera para o procedimento de transplante estão distribuídas em vários arquivos físicos que, no caso das fichas de pacientes transplantados, possuem dados gerais de cada paciente, de seus doadores, informações importantes para o transplante, o cirurgião que realizou o procedimento, além de informações sobre cada dia de pós-operatório desses pacientes. Já nas fichas dos pacientes que aguardam um transplante possui dados

gerais desses pacientes, dados imunológicos e uma série de exames que avaliarão a taxa de rejeição ou sobrevivência de um órgão no paciente.

Levando em consideração que atualmente o IMIP possui registros de mais de 1600 pacientes transplantados e mais de 500 pacientes aguardando na fila por um transplante, a realização de análises sobre esses dados torna-se muito custosa e complexa para um gestor realizar. A criação de um conjunto de dados abertos com esses registros beneficiará todos os envolvidos nas etapas do processo de realização de um transplante, pois auxiliará na visualização, geração de análises sobre esses dados, e realização de projetos de pesquisas.

As etapas de criação do *dataset* seguirão as etapas do ciclo de vida dos dados na Web, como descrito a seguir. A etapa de **Preparação** abrangerá um estudo e análise dos dados disponíveis sobre os pacientes de pré-transplante e transplantados. Nessa fase será necessário a digitalização dos documentos disponíveis e uma análise dos dados que são relevantes e dos dados que podem ser publicados em formato aberto, respeitando a privacidade de cada paciente, visto que muitos dados presentes nas fichas são sigilosos. A etapa de **Criação** diz respeito à modelagem dos conjuntos de dados e à criação do vocabulário para descrição dos dados. Nessa etapa utilizaremos uma ferramenta ou *api* para auxiliar na geração das triplas *RDF*, seguindo os princípios de *Linked Data.* Desta forma o dado estará aberto e conectado, possibilitando a realização de inferências e a criação de *links* com outros *datasets*.

Na etapa de **Avaliação** os especialistas do IMIP irão avaliar amostras dos dados para certificar a qualidade dos mesmos e de que nenhum dado infrinja a privacidade de cada paciente. Na etapa de **Publicação,** o conjunto de dados será disponibilizado em um portal de dados abertos que fará uso de um SGDW para o gerenciamento dos conjuntos de dados proposto em [Oliveira *et al.* 2018]. Esse SGDW é uma solução mais completa, pois permite a definição, criação, manutenção, manipulação e compartilhamento de conjuntos de dados na Web, enquanto que as soluções atualmente disponíveis se concentram mais na catalogação de conjuntos de dados. Publicaremos o *dataset* nos diversos níveis de abertura dos dados, seguindo os princípios de *Linked Data* e as melhores práticas para publicação de dados na Web.

Na etapa de **Consumo,** os dados estarão disponíveis em um portal para o consumo desses dados, a fim de que eles possam ser utilizados para a criação de visualizações, como gráficos, estatísticas, bem como para a realização de análises sobre os dados, como a quantidade de pacientes transplantados que precisaram ser readmitidos em um certo período de tempo após o procedimento de transplante. A etapa de **Feedback** é uma das etapas de suma importância no ciclo de vida dos dados, em especial para este trabalho, pois é nela que receberemos *feedback* sobre o conjunto de dados gerado de forma a mantê-lo em constante atualização e melhoria, bem como a escolha de novos dados a serem publicados. A última etapa é a de **Refinamento** e reflete o momento para correção de possíveis erros e, se for o caso, repetir todo o processo do ciclo de vida, desde a avaliação até um novo refinamento dos dados.

4. Metodologia

O processo de criação desse trabalho divide-se essencialmente em 4 etapas. A primeira etapa foi dedicada ao levantamento do estado da arte, onde encontramos alguns

conjuntos de dados relacionados à área de saúde e que estão em formato aberto e/ou conectado. A segunda etapa diz respeito ao estudo das boas práticas para publicação de dados, a fim de aplicá-las ao nosso projeto. A partir daí será possível a criação e publicação do *dataset* de acordo com essas boas práticas.

A terceira etapa desse trabalho é onde avaliaremos o *dataset* criado com o auxílio de especialistas na área. A quarta e última etapa é quando divulgaremos os resultados obtidos por meio da escrita de artigos e pela escrita da dissertação como requisito parcial para obtenção do título de Mestre.

5. Trabalhos Relacionados

Apesar do grande número de *datasets* com dados abertos e conectados, ainda são poucos na área de saúde, especificamente, em comparação com *datasets* sobre dados governamentais, por exemplo. Além disso, muitos dos *datasets* que contém dados sobre saúde estão em formato aberto, segundo a classificação de abertura dos dados proposto por Tim Berners-Lee, porém nem todos estão em formato conectado.

Além do site DataHub mencionado anteriormente, também podemos citar o site HealthData.gov[6] que incorpora 125 anos de dados de saúde nos EUA, onde podemos encontrar conjuntos de dados fornecidos por agências em todo o Governo Federal, bem como as ferramentas e aplicativos para manipulação e processamento de dados. É importante ressaltar que o DataHub possui conjuntos de dados abertos, porém nem todos estão conectados. Já o HealthData.gov possui conjuntos de dados conectados, porém nem todos são abertos.

Numa busca rápida no HealthData.gov, encontramos 2.737 *datasets* divididos entre os seguintes tópicos: *Health, State, National, Medicare, Hospital, Quality, Community, Inpatient*. Ao restringirmos a busca para *datasets* que possuam dados sobre "Transplant" ou "Transplantation" o resultado retorna apenas 7 conjuntos de dados[7] [8] [9] [10] [11] [12], tendo um deles encerrado. Destes 7 conjuntos de dados, podemos destacar 3 que possuem algumas características ou análises similares às que serão desenvolvidas neste trabalho.

O primeiro é o *dataset* "*Surgical Site Infections (SSIs) for Operative Procedures in Healthcare*[7]" que contém dados sobre infecções em locais cirúrgicos relatadas por um hospital para o Center for Disease Control and Prevention (CDC) e para o National Healthcare Safety Network (NHSN). O segundo *dataset* é o "*Incidence of Lung Transplants in the Medicare Population*[9]" com análises da incidência de transplantes de

[6] https://www.healthdata.gov/

[7] https://www.healthdata.gov/dataset/surgical-site-infections-ssis-operative-procedures-healthcare

[8] https://www.healthdata.gov/dataset/central-line-associated-bloodstream-infections-clabsi-healthcare

[9] https://www.healthdata.gov/dataset/incidence-lung-transplants-medicare-population

[10] https://www.healthdata.gov/dataset/medicare-ffs-30-day-readmission-rate-puf

[11] https://www.healthdata.gov/dataset/hcup-national-nationwide-inpatient-sample-nis-restricted-access-file

[12] https://www.healthdata.gov/dataset/hrsa-data-warehouse

pulmão na população do Medicare[13] nos últimos anos. O terceiro *dataset* é o "*Medicare FFS 30 Day Readmission Rate PUF[10]*" que contém dados da Taxa de Readmissão Hospitalar (PUF). Este dataset permite analisar as causas em que um beneficiário do Medicare é internado em um hospital dentro de 30 dias da data de alta após uma estadia anterior.

Além desses conjuntos de dados mencionados acima, podemos encontrar outros na literatura, porém eles tratam especificamente de mamografias, e/ou de algum tipo de câncer. Os poucos *datasets* que foram encontrados no filtro de "transplante" ou "doação de órgãos" não tratam especificamente do paciente em si, mas das causas possíveis de complicação e afins.

6. Considerações Parciais e Trabalhos Futuros

Este trabalho propõe a criação de um dataset aberto e conectado sobre os dados de pacientes dos transplantes realizados no IMIP. O *dataset* a ser disponibilizado caracteriza-se como importante fonte de informação para diversos novos estudos na área de saúde e afins, além de auxiliar na visualização e geração de análises sobre os dados publicados. Atualmente, o projeto encontra-se na fase de preparação do *dataset*. Considerando que os dados a serem publicados estão dispostos em diversos arquivos físicos, será necessário a inserção desses dados em uma base de dados para que então possamos analisar, juntamente com o médico responsável, quais dados podem estar em formato aberto, visto que muitos deles possuem informações particulares de cada paciente.

Referências

Abiteboul, S., Buneman, P., and Suciu, D. (2000). *Data on the Web: from relations to semistructured data and XML.* Morgan Kaufmann.

Berners-Lee, T., Connolly, D., and Swick, R. R. (1999). *Web architecture: Describing and exchanging data.* Disponível em: <https://www.w3.org/1999/04/WebData>. Acesso em: 27 mai. 2018.

Isotani, S. and Bittencourt, I. I. (2015). *Dados Abertos Conectados*. novatec, 1st edition.

Lóscio, B. F., Burle, C., and Calegari, N. (2017). *Data on the Web Best Practices*. Disponível em: <https://www.w3.org/TR/dwbp/>. Acesso em: 10 mai. 2018.

Lóscio, B. F., Oliveira, M. I. S., and Bittencourt, I. I. (2015). *Publicação e consumo de dados na web: Conceitos e desafios*. In *Minicurso SBBD*.

Oliveira, L. E. R. A., Oliveira, M. I. S., Santos, W. C. 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* (p. 2). ACM.

Pires, M. T. (2015). *Guia de Dados Abertos*.

W3C Escritório Brasil and Laboratório Brasileiro de Cultura Digital (2011). *Manual dos Dados Abertos: Desenvolvedores.*

[13] https://www.medicare.gov/

Versionamento de conjuntos de dados publicados na Web

Aluno: Wilker Cavalcante do Rego Santos

Email: wcrs@cin.ufpe.br

Orientadora: Bernadette Farias Lóscio

Email: bfl@cin.ufpe.br

Universidade Federal de Pernambuco - UFPE

Programa de Pós Graduação em Ciência da Computação – Centro de Informática

Nível: Mestrado

Ingresso: Março de 2017

Conclusão prevista: Fevereiro de 2019

Etapas Conluídas: Créditos em disciplinas, Referencial Bibliográfico, Definição do Problema, Especificação

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

Resumo. *A quantidade de dados publicados na Web vem crescendo, assim como esforços para encontrar melhores formas para publicação e consumo são constantes. Sabe-se que é importante que o consumidor de dados tenha acesso ao histórico de todas as versões dos conjuntos de dados que deseja consumir a fim de garantir uma melhor análise dos dados. Nesse contexto, este artigo tem como objetivo propor um modelo para o versionamento de conjuntos de dados publicados na Web, bem como propor uma solução para o armazenamento das versões a fim de ter a melhor velocidade de acesso, juntamente com o menor custo de armazenamento possível.*

1 Introdução

A Web surgiu como uma importante ferramenta para compartilhamento e consumo de informação. Com a popularização da Web, a quantidade de dados gerados por seus usuários cresce a cada dia. Em paralelo, novos paradigmas vem sendo desenvolvidos para encontrar novas formas de como os usuários podem fazer uso do que a Web oferece, fazendo desta área, de grande importância para a pesquisa.

Apesar da ideia de publicação de dados na Web não ser nova [3], apenas o ato de compartilhamento ainda não é o suficiente. Desta forma, o W3C (World Wide Web Consortium) vem trabalhando constantemente para encontrar melhores formas para padronizar a publicação e consumo de dados. Nesse sentido, foi criada uma recomendação composta por 35 boas práticas para dados na Web, com o intuito de facilitar a comunicação entre publicadores e consumidores de dados na Web [11].

Considerando que os conjuntos de dados podem mudar ao longo do tempo, [11] propõe que os mesmos sejam versionados para que os seus consumidores possam compreender como as mudanças estão alterando os dados e em que versão estão trabalhando. Porém, [11] também mostra que ainda não existe um consenso de quando uma nova versão do conjunto de dados será criada, ou que tipos de alterações poderão desencadear a criação de um novo conjunto de dados a partir do original. Por este motivo, este trabalho tem o objetivo de propor um modelo para o versionamento de conjuntos de dados publicados na Web. O modelo proposta visa guiar a comunidade científica para o entendimento e clareza sobre o que deve ser considerado para um adequado gerenciamento de versões dos conjuntos de dados publicados na Web.

É importante destacar que a forma como o armazenamento das múltiplas versões é realizada pode causar uma complexidade computacional desnecessária na recuperação dos dados, assim como um custo de armazenamento alto [10]. Assim, este trabalho também vem propor uma melhor forma de armazenar as versões dos conjuntos de dados de forma a viabilizar o melhor balanceamento entre velocidade de acesso e armazenamento, viabilizando, consequentemente, o consumo dos dados com base em histórico de versões.

O restante deste trabalho está organizado da seguinte forma. Na Seção 2, é apresentada a fundamentação teórica para o entendimento deste artigo. Na seção 3, a solução proposta é apresentada. Na Seção 4, a metodologia para o desenvolvimento do trabalho é detalhada. Na Seção 5, os trabalhos relacionados são discutidos. Por fim, na Seção 6 se tem a conclusão.

2 Fundamentação Teórica

Segundo [7], uma versão pode ser definida como um *snapshot* do estado de um objeto em um determinado momento, que pode refletir sua criação ou atualização. Mais especificamente, versionamento de dados pode ser determinado como uma técnica para proteger informações, de forma que, todas as informações de alterações realizadas ao longo do tempo nos objetos de dados possam ser extraídas do repositório.

Ainda de acordo com [7], em um sistema de versionamento deve ser possível salvar, representar e extrair versões. Sendo *salvar* o ato de armazenar o *snapshot* dos dados mantendo sua consistência. Quando uma versão é auto-contida e independente das outras versões no repositório, ela é *representada*. Por fim, quando a versão é *extraída*, ela tem um acesso rápido e fácil. Adicionando a isso, [7] também apresenta algumas técnicas para o versionamento de dados:

- *Split mirror*. Aqui, a inserção da nova versão fica agendada. Enquanto isso, uma cópia da imagem de todos os dados (chamada de "espelho") é salva no repositório constantemente. No momento da criação da versão, apenas as diferenças que estão armazenadas no repositório são recuperadas para que as alterações possam ser finalmente aplicadas.

- *Copy-old-while-update-new*. No início, a nova e a velha versão ocupam o mesmo espaço. Em seguida, a área afetada do objeto é salva no repositório antes da alteração. A outra parte não afetada do objeto também é copiada para o repositório de forma assíncrona.

- *Keep-old-and-create-new*. A nova versão é inserida em um espaço diferente, mantendo a antiga.

Sistemas de versionamento também podem ser classificados como *branching* e *non-branching* [7]. Sistemas de*Branching* utilizam o ato de realizar cópias de objetos, para que se possa efetuar alterações neles sem afetar os originais. Ao final, um merge pode ser realizado, ou seja, as alterações podem ser aplicadas nos objetos de origem. Sistemas *non-branching* preferem classificar suas alterações apenas com um identificador.

[10] relata um cenário em que uma grande quantidade de dados precisa ser versionada. De acordo com ele, soluções populares de Sistema de Controle de Versão como Git, SVN e Mercurial utilizam algoritmos muito simples e não são capazes de tratar conjuntos de dados muito grandes. Ele ainda demonstra a importância de encontrar um ponto de balanceamento entre velocidade de acesso e armazenamento.

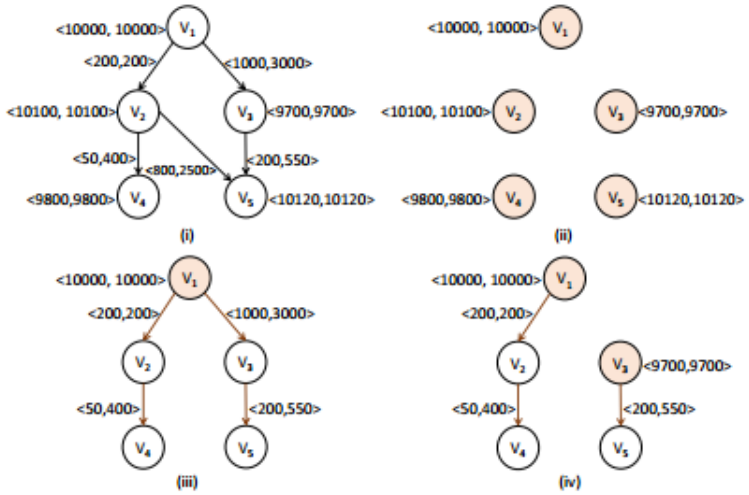


Figura 1: **Grafo de Versão [10]**

Na Figura 1(i) é possível observar um conjunto de dados com cinco versões, sendo V_1 a original. Derivados de V_1, temos duas versões diferentes: V_2 e V_3. Então, um *merge* é realizado envolvendo V_2 e V_3 originando V_5. E por último, assim como anteriormente, uma versão V_4 é criada a partir de V_2. Como pode ser visto, em V_1 são encontrado os valores $\langle 10000{,}10000 \rangle$, indicando que o custo de armazenamento e recriação é de 10000 quando guardado integralmente. Na aresta $V_1 \rightarrow V_3$, o valor é de $\langle 1000{,}3000 \rangle$, indicando que o custo para recriar V_3 a partir de V_1 é de 3000, enquanto o custo para armazenar os dados necessários para a recriação é de 1000. Nessa lógica, são apresentadas as Figuras 1(ii) e 1 (iii) que respectivamente mostram as mesmas versões quando guardadas integralmente e apenas as recriações. Como mostra [10], apesar da Figura 1 (ii) demonstrar um grafo em que todas as versões podem ser recuperadas individualmente, o custo de armazenamento é alto (49720). Enquanto que na Figura 1 (iii), que apenas as recriações são guardadas, existe um baixo custo de armazenamento (11450), porém com um alto custo para recuperação (13550, indo de $V_1 \rightarrow V_5$). Por fim, [10] ilustra na Figura 1 (iv) o que seria uma abordagem intermediária. Nesse caso, apenas as versões V_1 e V_3 são armazenadas totalmente enquanto as outras são recriadas no momento da recuperação. Desta forma, a Figura 1 (iv) mostra um custo de armazenamento entre os custos de (ii) e (iii) e um resultado satisfatório para a recuperação de V_2, V_4 e V_5.

3 Caracterização da Contribuição

Considerando as definições da literatura citadas anteriormente e seguindo as recomendações de melhores práticas de dados na Web [11], pode-se definir que uma nova versão para o conjunto de dados na Web deve ser gerada a partir de qualquer alteração, seja ela nos dados em si (i.e. inserção de novos dados), ou na estrutura (i.e. criação de uma nova propriedade). Para todas as versões, deve ser possível a recuperação total das mesmas, bem como os metadados estruturais e todos

os demais necessários para descreve-las. Novas versões também podem ter origem a partir de um subconjunto de dados ou a partir da união entre dois ou mais conjuntos.

3.1 *Branching* e *Merging*

Os diversos conjuntos de dados e suas versões podem interagir entre si ao longo do tempo. Uniões entre dados, criação de subconjuntos, correções e inserção de novos dados podem sempre ocorrer. Para o modelo proposto, é utilizado a técnica de *branching*, em que novos *branches* representam variações de um conjunto de dados, permitindo ao usuário realizar um *merge* ou não.

A Figura 2 representa um grafo de versão que contém três conjuntos de dados: A, B e C, sendo cada nó uma versão diferente e cada aresta um ponteiro para a próxima versão gerada. No início, apenas os conjuntos A e B existem em sua versão inicial nas *branches* *b* e *d* respectivamente. Logo mais, devido à alterações, novas versões dos dois conjuntos são criadas, originando A2 e B2. Depois, a partir deles são criadas as *branches*, *a* e *e*, frutos de variações (i.e. subconjuntos) tendo início em A2.1 e B2.1. É importante observar que, as respectivas *branches* originais de A e B continuam a gerar novas versões independentes de A2.1 e B2.1. Em um determinado momento, como fruto da junção entre A e B, o conjunto de dados C é formado, tendo assim sua própria *branch c*. Em determinado momento, um *merge* é feito de A2.2 para a *branch* *b* e a versão A5 é criada. Para este modelo, deve-se levar em consideração as seguintes regras:

- *Regra 1.* Quaisquer alterações nos conjunto de dados, seja em sua estrutura, nos metadados ou nos dados em si, devem gerar uma nova versão.
- *Regra 2.* A criação de subconjuntos de dados deve gerar uma nova versão em uma *branch* diferente.
- *Regra 3.* A combinação de conjuntos de dados com outros que não pertençam ao domínio em questão deve gerar uma nova versão em uma *branch* diferente.
- *Regra 4. Merges* são permitidos apenas para as *branches* que têm a origem em comum.
- *Regra 5.* Combinação entre conjuntos de dados deve gerar um novo conjunto de dados com sua própria *branch*.

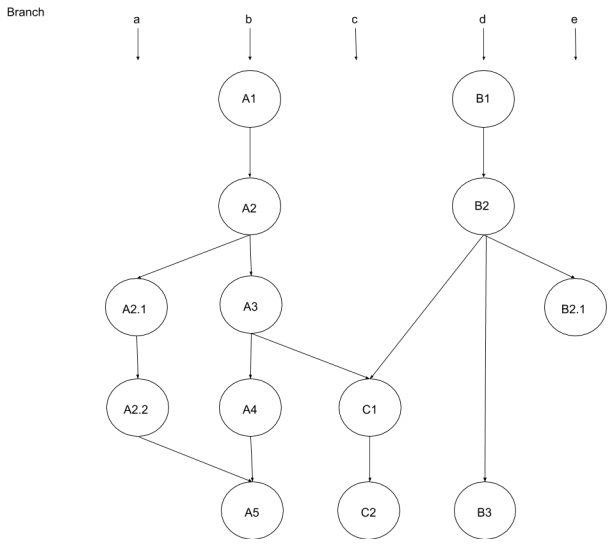


Figura 2: **Grafo de Versão de dados na Web**

3.2 Identificador de versão

As versões precisam de um identificador único [11]. Além disso, é importante que o número de identificação transmita o que uma versão representa dentre todas as outras. Para isso, o número deve refletir quantas alterações foram realizadas desde a criação, que tipo de alterações e em que *branch* está. Tudo isso de maneira que tanto seres humanos quanto máquinas possam interpretar. A Figura 3 demonstra o modelo proposto.

$$V.S\_d.e.c$$

Figura 3: **Modelo de Identificador de Versão**

O primeiro número, representado pelo V indica o numero de versão maior, ou seja, a versão da *branch* original. O número de subversão, ilustrado como S representa o número de versão de uma *branch* derivada de outra. Quanto mais *branches* criadas, maior devem ser o número de casas S. Os demais, d, e e c, são respectivamente quantidade de alterações em dados, estrutura e correção de erros, que devem ser somados quando as modificações referentes são realizadas. Utilizando como exemplo o conjunto de dados A da Figura 2, para representar a primeira versão A1, o número de versão deve ser 1.0_0.0.0. Partindo para A2, supondo que houve uma mudança na estrutura, o número deve ficar 2.0_0.1.0. Já para A2.1, a versão deve ser identificada por 2.1_0.0.0. É importante observar que os valores referentes à alterações de dados, estrutura e correções devem ser anuladas quando uma nova *branch* é criada (i.e supondo que B2 tenha o número de versão 2.0_1.0.0, B1.2 deve ficar 2.1_0.0.0).

3.3 Armazenamento das Versões

Quando se trada do armazenamento das versões dos conjuntos de dados, o modelo proposto considera o armazenamento total de uma versão apenas quando esta representa a criação de um novo conjunto de dados, uma nova *branch*, a realização de um *merge* ou quando ocorrer uma alteração muito grande que exija uma grande complexidade computacional significativa para a reprodução. Nos outros casos, apenas a diferença entre as versões deve ser guardada, permitindo a recriação das mesmas a partir de uma original.

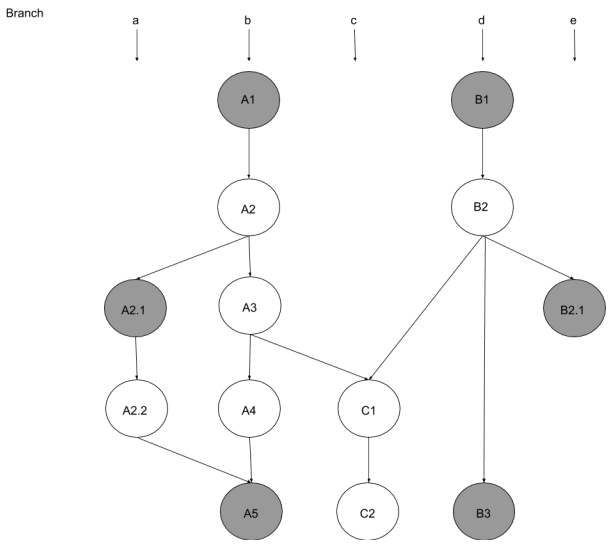


Figura 4: **Armazenamento em Grafo de Versão**

A Figura 4 ilustra o grafo apresentado anteriormente, desta vez dando destaque às versões que devem ser armazenadas. Os nós A1 e B1 devem ser armazenados por se tratarem de versões originais, de onde as oriundas devem se basear para a recriação. As versões A2.1 e B2.1 por serem

de outras *branches*, devem ser armazenadas também. Em A5 um *merge* é feito, então como versão fruto, também deve ser armazenada. Com a origem em B2, supondo que uma grande alteração (i.e uma grande quantidade de novos dados) foi adicionada, B3 deve ser armazenada integralmente. Não existe uma regra que defina quando nova versão deve ser armazenada completamente devido a sua complexidade, cabe ao usuário analisar seus recursos computacionais e encontrar o melhor caso de uso.

4 Metodologia

Para a realização deste trabalho, as seguintes etapas foram definidas:

1. **Análise literária**. Nessa etapa, uma análise da literatura foi realizada a fim de encontrar trabalhos semelhantes. Trabalhos relacionados à armazenamento de dados e versionamento foram levados em consideração.
2. **Implementação**. Em um ambiente controlado, as possíveis soluções propostas devem ser implementadas e testadas. Nesta fase, o foco deverá ser a medição do desempenho computacional.
3. **Experimento**. Nessa etapa, o método de versionamento será utilizado nos conjuntos de dados publicados no portal de dados abertos da Universidade Federal de Pernambuco.
4. **Divulgação dos resultados**. Por fim, todos os resultados obtidos devem ser divulgados através de publicações de artigos em outros anais e a escrita da dissertação de mestrado.

5 Trabalhos relacionados

Com relação a um modelo para versionamento de conjuntos de dados na Web, não existem propostas na literatura que se posicionem em relação ao assunto. Podem ser encontradas abordagens semelhantes, como a de versionamento de ontologias [5], versionamento de objetos em bancos de dados [1] e versionamento de documentos XML [4]. Também é um objeto de estudo a recuperação de *snapshots* em de Banco de Dados orientados a vetores [8]. Também existem trabalhos publicados na área de Bancos de Dados temporais [2], em que é possível a recuperação de uma versão em um determinado espaço de tempo.

Em relação à questão de armazenamento de conjuntos de dados, [9] apresenta o DATAHUB, uma solução que permite ao usuário, salvar, criar *branches*, realizar *merges* em grandes conjuntos de dados. Além disso, algoritmos apresentados em [10] podem ser utilizados para solucionar questões como o balanceamento entre armazenamento e custo de recuperação. Por fim, em [6] é apresentado um método para armazenamento em que a compressão dos arquivos elimina custos desnecessários.

6 Considerações e trabalhos futuros

Este artigo apresentou um modelo para o versionamento de conjuntos de dados na Web. Atualmente, um levantamento na literatura ainda está sendo realizado em busca de responder se o modelo proposto realmente atende aos consumidores de dados e se existem outras pendências na área que possam ser solucionadas. Além disso, experimentos devem ser feitos como parte do processo de evolução da abordagem do armazenamento dos conjuntos de dados.

Referências

[1] David Beech e Brom Mahbod. "Generalized version control in an object-oriented database". Em: *Data Engineering, 1988. Proceedings. Fourth International Conference on*. IEEE. 1988, pp. 14–22.

[2] Abdullah Uz Tansel et al. *Temporal databases: theory, design, and implementation*. Benjamin-Cummings Publishing Co., Inc., 1993.

[3] Tim Berners-Lee, Dan Connolly e Ralph R Swick. “Web architecture: Describing and exchanging data”. Em: *W3C Note, June* (1999).

[4] Shu Yao Chien, Vassilis J Tsotras e Carlo Zaniolo. “XML document versioning”. Em: *ACM SIGMOD Record* 30.3 (2001), pp. 46–53.

[5] Michel CA Klein e Dieter Fensel. “Ontology versioning on the Semantic Web.” Em: *SWWS*. 2001, pp. 75–91.

[6] Sean Quinlan e Sean Dorward. “Venti: A New Approach to Archival Storage.” Em: *FAST*. Vol. 2. 2002, pp. 89–101.

[7] Ningning Zhu. *Data Versioning Systems*. Rel. téc. Stony Brook University, 2003.

[8] Emad Soroush e Magdalena Balazinska. “Time travel in a scientific array database”. Em: *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE. 2013, pp. 98–109.

[9] Anant Bhardwaj et al. “Datahub: Collaborative data science & dataset version management at scale”. Em: *arXiv preprint arXiv:1409.0798* (2014).

[10] Souvik Bhattacherjee et al. “Principles of dataset versioning: Exploring the recreation/storage tradeoff”. Em: *Proceedings of the VLDB Endowment* 8.12 (2015), pp. 1346–1357.

[11] Newton Calegari Bernadette Farias Lóscio Caroline Burle. *Data on the Web Best Practices*. URL: `https://www.w3.org/TR/dwbp/`. (accessed: 05.06.2018).

Caracterização e Comparação de Campanhas Promovendo o Outubro Rosa e o Novembro Azul no Twitter

Roberto Walter[1], **Karin Becker**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

roberto.wtr@gmail.com, karin.becker@inf.ufrgs.br

Nível: Mestrado
Mês e ano de ingresso: Março de 2017
Mês e ano previstos para defesa: Março de 2019
Etapas Concluídas: Revisão Bibliográfica Preliminar; Definição da Abordagem; Experimentos Iniciais

Abstract. *The Pink October and Blue November campaigns seek population education and awareness regarding breast and prostate cancers. Given the success of the Pink October campaigns and the low male engagement in the Blue November, in this article we present a first comparative evaluation of these campaigns in the online context. Considering Twitter as a platform that provides community engagement, we study the demographic characterization of gender and age, geographic location, activity periods, and user engagement with campaign-related tweets. We found and mapped differences and similarities of strategies in campaigns as well as differences in characteristics of users engaged in campaigns. We have also identified that online campaigns on Twitter have reached the target audience of the campaigns, which are men and women over 40 years of age. Finally, we discussed the campaign promoters and their impact on the social network.*

1. Introdução

Estudos estimam que uma em cada oito mulheres desenvolverá o câncer de mama durante sua vida [Altekruse et al. 2010]. A campanha do Outubro Rosa (OR) tem anualmente focado em aumentar a participação feminina em exames de detecção do câncer em seus estágios iniciais [Jacobsen and Jacobsen 2011]. Estes esforços têm obtido êxito, com aumento da participação em mais de 50 pontos percentuais entre 1987 e 2010 [for Disease Control and Prevention 2012]. No entanto, esse desenvolvimento não é observado na campanha do Novembro Azul (NA) [Glynn et al. 2011].

Trabalhos têm examinado a importância do uso de mídias sociais para propagar o conhecimento sobre a saúde pública [Bravo and Hoffman-Goetz 2016]. Estudos relacionados a campanhas no Twitter contra o câncer se concentram no acompanhamento do volume de mensagens ao longo do tempo [Nastasi et al. 2017, Jacobson and Mascaro 2016] para as campanhas OR e NA, e na investigação da relação entre o conteúdo e a campanha (e.g. diferenças pelo tipo de câncer [Borgmann et al. 2016], relação dos tópicos e as metas do OR [Thackeray et al. 2013], estratégias de angariação de fundos para o NA [Prasetyo et al. 2015]).

No entanto, não foram encontrados estudos mais aprofundados sobre o NA, que englobem uma avaliação de atividade durante o mês de campanha, de caracterização e localização dos perfis, do engajamento dos usuários, ou mesmo uma comparação da campanha NA com a OR. Há uma grande oportunidade de realizar uma avaliação do caso de sucesso do OR, que expandiu os seus números desde o início das campanhas, e compará-lo ao NA, que tem apresentado resultados tímidos sobre o engajamento masculino. Assim, seria possível averiguar se existem divergências relevantes nos padrões de desenvolvimento das campanhas, e identificar se o público alvo do NA está sendo atingido da mesma forma como está sendo o público alvo do OR.

O objetivo deste trabalho é permitir um melhor conhecimento sobre as campanhas *online* do NA, caracterizando-as e comparando-as às campanhas do OR. Assim, buscamos responder as seguintes perguntas sobre o NA, em comparação com o OR:
QP 1: Os usuários apresentam características demográfico e geográfico similares?
QP 2: As campanhas apresentam características temporais similares?
QP 3: As campanhas apresentam abrangência similar na rede social?
QP 4: Os tópicos de conteúdo das campanhas são semelhantes?
QP 5: Os padrões das campanhas permanecem ao longo dos anos?

Este estudo constitui uma experiência pioneira comparando campanhas de câncer no Twitter, com as seguintes contribuições:

- Complementamos estudos anteriores (e.g. [Prasetyo et al. 2015], [Nastasi et al. 2017, Jacobson and Mascaro 2016], [Thackeray et al. 2013]) com uma comparação entre OR e NA, e análises adicionais: i) do público envolvido em cada campanha, verificando se corresponde ao respectivo público-alvo; ii) dos padrões temporais de atividades, detectando diferenças/semelhanças por campanha e país; iii) das diferenças na cobertura de *tweets*, mostrando que o NA tem alcance limitado e que organizações e celebridades não desempenham um papel proeminente. Essas análises podem ser estendidas a outras campanhas sobre câncer.
- Propomos métricas para uma campanha com resultados positivos (OR) e a contrastamos com uma campanha similar (NA) com menos engajamento. Essa comparação ajuda a entender os fatores que influenciam o alcance da campanha do

Tabela 1. Visão geral de dados e métodos de trabalhos relacionados.

Obra	Dados		Contexto	Dados Temporáis	Engajamento	Tipo de Usuário	Demografia			Geolocalização	
	Origem	Integração					Idade	Sexo	Técnica Definição	Estado Político	Técnica Definição
[ElSherief et al. 2017]	Twitter	API Twitter	Violência baseada no gênero	-	Sim	-	Sim	Sim	Face ++	-	-
[Olteanu et al. 2016]	Twitter	API Twitter	Equidade Racial	Sim	-	Sim	Sim	Sim	Crowdworkers	-	-
[Lotan et al. 2011]	Twitter	API Twitter	Primavera Árabe	Sim	Sim	Sim	-	-	-	País	Auto-relatado
[Glynn et al. 2011]	Google	Google Insights for Search	Outubro Rosa	Sim	-	-	-	-	-	-	-
[Thackeray et al. 2013]	Twitter	API Twitter	Outubro Rosa	Sim	Sim	Sim	-	-	-	-	-
[Borgmann et al. 2016]	Twitter	Symplur, Tweet Archivist, Twitonomy	Oncologia Urológica	Sim	Sim	Sim	-	-	-	País	Twitonomy
[Nastasi et al. 2017]	Twitter	Symplur	Outubro Rosa	-	Sim	Sim	-	-	-	País/Continente	-
[Prasetyo et al. 2015]	Twitter	API Twitter	Novembro Azul	-	Sim	Sim	-	-	-	País/Continente	Auto-relatado
[Jacobson and Mascaro 2016]	Twitter	API Twitter	Novembro Azul	Sim	Sim	-	-	Sim	-	-	-

NA no Twitter e, consequentemente, aumenta o engajamento da população-alvo em exames de detecção precoce do câncer.

- Estabelecemos um *baseline* para essas campanhas, monitoramos seus padrões no tempo. Isto também permite acompanhar a evolução destes padrões no ano futuro.

2. Trabalhos Relacionados

Vários são os trabalhos que estudam o perfil de engajamento de usuários do Twitter em diferentes causas. A Tabela 1 lista alguns trabalhos com seu contexto de aplicação, e desafios comuns a todos: coleta/integração de dados, avaliação de dados temporais, caracterização e engajamento de usuários, determinações demográficas, e geolocalização.

Estes trabalhos buscam criar um *dataset* do contexto estudado, usando principalmente a API do Twitter. As análises temporais em geral são avaliações de ativismo dos usuários. O engajamento é medido pela atividade de *tweets* e *retweets*, detalhados por informações demográficas, geolocalização ou categorização de usuários. Assim, busca-se definir as características de perfis que possam ter influência/participação nas mobilizações. Contudo, a definição dessas informações são um problema, porque ao capturar o perfil de um usuário o Twitter não solicita esses dados, e permite que a localização geográfica seja informada em um campo aberto, que muitas vezes são inválidas. Recursos como o Face++, Google Maps, ou plataformas coletivas de definição de dados têm sido utilizadas para buscar uma definição destes dados do usuário com informações que o mesmo possa ter informado em seu perfil, como por exemplo a imagem de perfil.

Entre os trabalhos relacionados que analisam campanhas relacionadas a câncer, alguns desenvolveram indicadores do OR sobre atividade, engajamento, tópicos abordados e categorias de usuários [Thackeray et al. 2013, Nastasi et al. 2017]. Diferenças nos tópicos de tweets das campanhas por tipo de câncer foram examinadas em [Borgmann et al. 2016]. Uma análise das atividades de angariação de fundos relacionadas com NA em diferentes países foi desenvolvida em [Prasetyo et al. 2015]. Não foram encontrados estudos mais aprofundados sobre o NA, que englobem uma avaliação de atividade durante o mês de campanha, de caracterização do perfil dos participantes e de seu engajamento, nem que o comparem à campanha OR, apesar de suas similaridades.

3. Dados e Métodos

Com base nos trabalhos relacionados apresentados na seção anterior, identificam-se diferentes abordagens para análise de campanhas no Twitter. As análises consideram informações para identificação do perfil dos usuários, análises de atividades para mapear os períodos de geração de conteúdo e o impacto das campanhas nas mídias sociais. Os dados de *tweets* e perfil dos usuários foram coletados do Twitter a partir da API de consulta aos dados públicos. A consulta foi efetuada com um conjunto inicial de *hashtags* que foram levantadas pelos autores como relacionadas às campanhas do OR e NA nos

anos anteriores, e ao examinar as relações no Twitter, agregou-se um novo conjunto de *hashtags*. A partir desses dados, os métodos de tratamento dos dados, obtenção de novos dados e desenvolvimento das análises são desenvolvidos da seguinte forma:

Avaliação de características temporais: Para coleta dos dados, definimos inicialmente o período do mês das campanhas (outubro e novembro), precedido e sucedido de uma semana (24/Set/2017 a 07/Dez/2017). Isso nos permite avaliar a distribuição da frequência dos *tweets*, e identificar se as campanhas possuem mais enfoque em determinados períodos. Realizamos também uma separação dessas avaliações por país e comparamos as duas campanhas, assim identificamos se condições culturais ou políticas afetam as campanhas. Em uma etapa posterior, estenderemos a dados de outros anos.

Caracterização demográfica e geográfica: O Twitter não solicita informações demográficas para seus usuários, e a localidade é um campo aberto sem validação. Utilizamos a ferramenta Face++ para definir a idade e gênero a partir da foto do perfil. O Google Maps[1] foi utilizado para obtenção do país a partir da informação de localidade no perfil (apenas 0.33% dos *tweets* coletados são georreferenciados). Criamos a categoria do usuário como Celebridade, Organização, ou Indivíduo. O perfil de Celebridade deve ser verificado pelo Twitter, possuir mais de 100 mil seguidores, e um gênero identificado pelo Face++ [Thackeray et al. 2013]. Organização é um perfil verificado pelo Twitter e que não possui o gênero identificado. Os demais são classificados como Indivíduo. Com essas informações, aplicamos uma avaliação demográfica e geográfica das campanhas.

Definição e identificação da abrangência das campanhas na rede social: Para avaliação da abrangência e engajamento das campanhas, assim como em [Thackeray et al. 2013] levantamos algumas estatísticas como o nº de usuários envolvidos, nº de *tweets*, e média de *tweets* por usuário. Além disso, realizamos uma comparação das estruturas de conexão dos usuários via *retweets*, avaliando a largura e a profundidade das conexões. Isso nos permite ter uma percepção da propagação dos *tweets* na rede.

Definição e identificação do conteúdo abordado: A avaliação de conteúdo nos permite identificar se as campanhas possuem tópicos similares. Para isso, identificaremos os tópicos com LDA [Blei et al. 2003] e os categorizaremos com base em um conjunto de palavras chaves. Essa abordagem foi aplicada nas campanhas no Twitter do OR [Thackeray et al. 2013], no entanto, vamos utilizar esse método também no NA. Para isso, o conjunto de palavras chaves utilizado para a categorização do OR sofrerá algumas adaptações específicas para comportar os tópicos do NA. Por exemplo, *pink* e *mastectomy* serão trocadas por *blue* e *prostatectomy*. Para a definição dos tópicos com LDA, o conjunto de *tweets* passará primeiramente por etapas de remoção de *stop words* e *stemming*.

Os trabalhos geralmente aplicam algumas dessas etapas para uma campanha. Propomos atingir as diferentes abordagens, e compará-las entre o OR e o NA. Isso permitirá identificar se os resultados do NA são um reflexo de sua abordagem. Entendemos que a campanha do NA é mais recente e isso é um fator, mas também que é importante avaliá-la e compará-la com uma campanha bem consolidada, para saber se ela precisa de adequações para melhorar seus resultados.

4. Resultados Preliminares

Nesta seção são apresentados os experimentos para as três primeiras questões de pesquisa (QP1-QP3) definidas na Seção 1. Estes resultados estão detalhados em um artigo (com-

[1]https://developers.google.com/maps/ Acesso em: 5 de julho de 2018

pleto) aceito no SBBD 2018 [Walter and Becker 2018].

4.1. QP 1: Os usuários apresentam características demográfico e geográfico similares?

Identificamos que cada gênero engaja mais na campanha da qual é alvo, isto é, mulheres engajam mais no OR do que no NA, e homens engajam mais no NA do que no OR. Observamos também que nas duas campanhas o gênero que mais engaja é o masculino, com 62.7% da participação no NA e 49.09% no OR. Há também o grupo das organizações, com participação similar em ambas campanhas (7.26% no NA e 5.63% no OR).

Verificamos maior participação do grupo etário 41+, alvo da campanha, com 62.46% no OR e 57.95% no NA. Comparamos esta participação com *tweets* de propósito geral, onde apenas 4.8% dos participantes são do grupo 41+ [Sloan et al. 2015], e concluímos que ambas campanhas atingem seus públicos alvos.

Na avaliação dos países, nos quatro com o maior número de *tweets* sobre as campanhas (EUA, Reino Unido, Canadá e Brasil), o OR atrai mais participação em relação ao NA, com uma pequena diferença no Canadá (3.11%), e maior nos EUA (mais de 90%).

Conclui-se assim que ambas as campanhas atingem seu público alvo, no tocante à gênero e faixa etária, mas que os níveis de consciência e engajamento são afetados pela cultura ou políticas próprias a cada país.

4.2. QP 2: As campanhas apresentam características temporais similares?

A distribuição por data de postagem (Fig. 1) indica que a quantidade de *tweets* nos dias que precedem as campanhas são baixos, comparados ao período oficial. As campanhas apresentaram um pico no início do seu mês. Utilizando Normalized Cross-Correlation, observamos que Canadá, Reino Unido e Estados Unidos apresentam atividades temporais similares nas duas campanhas, com correlações acima de 0.90 no NA e 0.70 no OR. O Brasil não possui correlações fortes com os países.

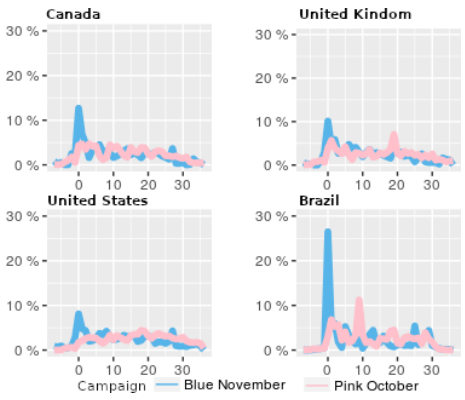


Fig. 1. Distribuição países

Identifica-se que alguns países possuem atividades similares. Em geral, a campanha do OR apresentou mais regularidade de atividades, e o NA teve oscilações de forma mais ou menos acentuada em cada um dos países.

4.3. QP 3: As campanhas apresentam abrangência similar na rede social?

Identificamos que o número de usuários, *tweets*, e a média de *tweets* por usuário é maior na campanha do OR para todas as categorias de usuários. Isto indica que, além da maior participação de usuários, em média os usuários envolvidos na campanha do OR se envolvem com mais postagens, comparados aos participantes no NA.

Tweets podem ser retuitados e atingir uma rede maior do que os seguidores imediatos do perfil que o postou, assim buscou-se compreender as relações por meio dos *retweets* entre os usuários. Para cada campanha construímos um grafo da estrutura das relações, e observamos que o nível de conexão entre os usuários do OR é maior que no NA, com os usuários compartilhando bem mais informação de um modo geral.

Através de uma avaliação de frequência de *retweets* por data de criação (Fig. 2), identificamos um maior número relacionado ao OR, onde a quantidade de *retweets* fica em torno de 300 em diversas datas, além de um número maior de *tweets* retuitados por dia. No NA essa máxima de *retweets* fica um pouco acima de 50, ou seja, 16,6% do número do OR. Verifica-se também que a mediana é mais alta no OR (8 *retweets*), comparada ao NA (4). Esta superioridade também é verificada no indicador do 3º quartil, onde são observados 34 *retweets* para o OR, e 8 para o NA.

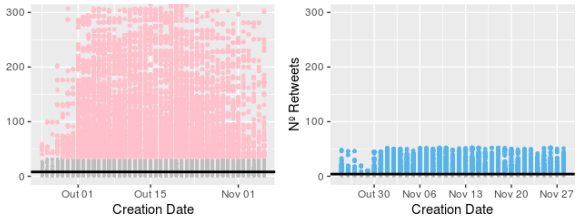


Fig. 2. Retweets por data de criação

Na distribuição do grau de conexões a partir de *retweets*, foi identificado que a maioria dos usuários são retuitados poucas vezes, enquanto pouquíssimos usuários possuem várias conexões por *retweets*. Conclui-se que a abrangência de propagação dos *tweets* através de *retweets* é maior para o OR, pelo fato dos graus de *retweets* serem mais frequentes do que o NA, e os *tweets* atingirem graus de *retweets* que o NA não atinge.

Conclui-se que as campanhas não apresentam abrangência similar. Além de mais numerosos, os usuários do OR possuem maior média de *tweets*. A propagação destes *tweets* também é maior no OR, pelo fato do engajamento através de *retweets* ser maior em quantidade de *retweets*, frequência, e no tamanho das conexões de usuários através de *retweets* entre si. Isto se aplica também em cada uma das categorias dos usuários, indicando que a influência de todas as categorias dos usuários é mais forte no OR.

5. Próximas Etapas e Futuras Publicações

Atualmente estamos expandindo as análises para as duas últimas questões de pesquisa: a) avaliação dos tópicos expressos nos tweets *tweets* (QP4) e b) um estudo sobre estes padrões ao longo dos anos (QP5). Quanto à análise dos tópicos, o trabalho de [Thackeray et al. 2013] utilizou LDA para definição dos tópicos e fez uma categorização dos mesmos com base em palavras chaves para a campanha do OR. Utilizamos este trabalho como um *baseline* para extração e interpretação dos tópicos do OR e NA. Os resultados foram comparados para identificar se as campanhas possuem categorias de conteúdos semelhantes ou diferentes, e os resultados preliminares foram submetido para a conferência Web Intelligence 2018. Para melhorar a interpretação dos agrupamentos resultantes do LDA, e comparação dos tópicos nas duas campanhas, atualmente estamos trabalhando em um método baseado na semelhança de *word embeddings*.

Estamos também gerando uma base de dados contendo *tweets* de anos anteriores a 2017, e pretendemos coletar os dados de 2018. Essas nova bases permitirão confirmar se os padrões encontrados se confirmam ao longo do tempo, bem como uma investigação da evolução dos tópicos. Com estes resultados completamos nosso trabalho com contribuições na área social com a avaliação de campanhas sobre a saúde pública, e na área de computação com uma análise de tópicos ao longo dos anos combinando LDA com *word embeddings*. Pretendemos publicar estes resultados completos em um *journal* conceituado da área de computação, ainda a ser definido.

References

Altekruse, S., Kosary, C., Krapcho, M., Neyman, N., Aminou, R., Waldron, W., Ruhl, J., Howlader, N., Tatalovich, Z., Cho, H., et al. (2010). Seer cancer statistics review,

1975–2007. *Bethesda, MD: National Cancer Institute*, 7.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Borgmann, H., Loeb, S., Salem, J., Thomas, C., Haferkamp, A., Murphy, D. G., and Tsaur, I. (2016). Activity, content, contributors, and influencers of the twitter discussion on urologic oncology. In *Urologic Oncology: Seminars and Original Investigations*, volume 34, pages 377–383. Elsevier.

Bravo, C. A. and Hoffman-Goetz, L. (2016). Tweeting about prostate and testicular cancers: Do twitter conversations and the 2013 movember canada campaign objectives align? *Journal of Cancer Education*, 31(2):236–243.

ElSherief, M., Belding, E. M., and Nguyen, D. (2017). # notokay: Understanding gender-based violence in social media. In *ICWSM*, pages 52–61.

for Disease Control, C. and Prevention (2012).

Glynn, R. W., Kelly, J. C., Coffey, N., Sweeney, K. J., and Kerin, M. J. (2011). The effect of breast cancer awareness month on internet search activity-a comparison with awareness campaigns for lung and prostate cancer. *BMC cancer*, 11(1):442.

Jacobsen, G. D. and Jacobsen, K. H. (2011). Health awareness campaigns and diagnosis rates: evidence from national breast cancer awareness month. *Journal of health economics*, 30(1):55–61.

Jacobson, J. and Mascaro, C. (2016). Movember: Twitter conversations of a hairy social movement. *Social Media+ Society*, 2(2):2056305116637103.

Lotan, G., Graeff, E., Ananny, M., Gaffney, D., Pearce, I., et al. (2011). The arab spring—the revolutions were tweeted: Information flows during the 2011 tunisian and egyptian revolutions. *International journal of communication*, 5:31.

Nastasi, A., Bryant, T., Canner, J. K., Dredze, M., Camp, M. S., and Nagarajan, N. (2017). Breast cancer screening and social media: a content analysis of evidence use and guideline opinions on twitter. *Journal of Cancer Education*, pages 1–8.

Olteanu, A., Weber, I., and Gatica-Perez, D. (2016). Characterizing the demographics behind the# blacklivesmatter movement. *OSSM. http://arxiv. org/abs/1512.05671*.

Prasetyo, N. D., Hauff, C., Nguyen, D., van den Broek, T., and Hiemstra, D. (2015). On the impact of twitter-based health campaigns: A cross-country analysis of movember. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pages 55–63.

Sloan, L., Morgan, J., Burnap, P., and Williams, M. (2015). Who tweets? deriving the demographic characteristics of age, occupation and social class from twitter user metadata. *PloS one*, 10(3):e0115545.

Thackeray, R., Burton, S. H., Giraud-Carrier, C., Rollins, S., and Draper, C. R. (2013). Using twitter for breast cancer prevention: an analysis of breast cancer awareness month. *BMC cancer*, 13(1):508.

Walter, R. and Becker, K. (2018). Caracterização e comparação das campanhas do outubro rosa e novembro azul no twitter. In *Simpósio Brasileiro de Banco de Dados 2018*.

A Framework for Identification and Monitoring of Profiles and Behaviors of Users Based on Mobile App Usage

Nielsen Luiz Rechia Machado[1], **Duncan Dubugras Alcoba Ruiz**[1]

[1]School of Technology, Pontifical Catholic University of Rio Grande do Sul - PUCRS
Computer Science Ph.D. Program, 90619-900 – Porto Alegre – RS – Brazil

nielsen.machado@acad.pucrs.br, duncan.ruiz@pucrs.br

Level: Ph.D.
Enrollment in the program: March 2015
Proposal's defense date: July 2017
Conclusion expected to: March 2019
Concluded steps: Credits; Bibliographic revision; Qualifying exam; Implementation of first Framework prototype, Analyses of experimental results;
Future steps: Improve the framework; Perform new experimental results, Evaluate the framework, Write thesis; Ph.D. Defense.
Publications: [Machado and Ruiz 2017]

Resumo. *Nos último anos, o uso de dispositivos móveis, bem como de seus aplicativos (apps) cresceu significativamente. Além disso, a disputa de mercado faz com que empresas deste ramo foquem na fidelização de seus clientes. Estes clientes realizam diariamente muitas atividades por meio de apps, o que gera, em tempo real, uma grande quantidade de eventos. Diante disso, a identificação e o monitoramento de perfis e comportamento de tais clientes podem contribuir para minimizar situações de risco, como a perda destes clientes. Assim, esta pesquisa propõe um framework para identificação e monitoramento de perfis e comportamentos de uso de apps em dispositivos móveis por usuários, visando segmentá-los em perfis de uso, para identificar comportamentos infrequentes que representem situações de risco para fabricantes de dispositivos móveis.*

Abstract. *Over the last years, the use of mobile devices and applications (apps) significantly grow. In addition, the technological innovation and fierce dispute to conquer the mobile market make companies increase their attention to their clients' loyalty. These clients perform daily many activities through apps generating a large number of events in real time. In this sense, the identification and monitoring of profiles and behavior of these clients can contribute to minimize risk situations, such as turnover. Therefore, this study proposes a new framework for the identification and monitoring of profiles and behaviors based on users' app usage. It aims to segment users into profiles seeking to identify infrequent behaviors that could represent risk situations for mobile device manufacturers.*

1. Introduction and Background

Mobile phones have evolved from simple communication devices to dynamic tools that provide advanced functionality to assist users in their daily activities [Hamka et al. 2014].

People use several applications (*apps*) for a variety of goals, such as read books, watch videos, take pictures, and so on. The number of apps available between 2012 and 2017 grew five times (i.e., from 675,000 to 3,300,000)[1]. The considerable amount of data issued by the use of apps can be categorized as a Data Stream (DS). Gama [Gama 2010] defines a DS as stochastic processes in which events occur continuously and independently of each other. Such data usually have large Volume (Big Data), large Variety (different kinds of data), and are produced at high Velocity (ongoing and in real time). Further, it should be collected from a considerable amount of users in real time. In a DS scenario, it is important to investigate the changes in the data distribution, namely Concept Drift. Indeed, new concepts arise and known concepts may evolve or disappear, which are discovered applying Novelty Detection techniques [Gama 2010]. In this sense, an effective tool, able to perform information analysis and capable of helping researchers and companies to extract knowledge about these app DSs, is almost mandatory.

Advances in mobile device industries, such as new services and technologies, as well as advances in the areas of data mining and machine learning, increased the competition in the market. Mobile device companies seek to keep their customers loyal and engaged because they know that the cost of attracting a new customer is six times greater than the amount spent to retain the old customers. Thus, these companies are under intense pressure to identify and monitor customers' behaviors. Since customers are the "source material", and the market is saturated, new methods for the management of these customers are vital for the survival and development of such companies [Almana et al. 2014]. However, tasks to identify app usage profiles and to monitor customers' behaviors are difficult. Moreover, it is still difficult to access data from mobile devices as app usage. The first mobile datasets were made available only in the last few years [Wagner et al. 2013]. Nevertheless, even considering the main datasets for extracting information from customers, such data have few or no detailed information of app usage and are usually private or not available due to privacy-preserving policies.

The main objective of this research is *to develop a framework for identification and monitoring of profiles and behaviors of users based on app usage on mobile devices*. We aim to segment the users into usage profiles seeking to identify infrequent behaviors that could represent risk situations for mobile device manufacturers. Specifically, our proposed framework is designed to (a) deal with different types of data from mobile app usage, (b) identify usage patterns, (c) group users in usage profiles, and (d) monitor profiles and behaviors of users aiming to segment them according to their variations through time. In a Computer Science view, we are developing an application for mining and monitoring app usage DSs that may have an impact on the mobile device industry.

2. Related Work

We carried out a Systematic Review [Kitchenham 2004] of literature to investigate studies seeking to identification and the monitoring of usage profiles. This review explores 20 related work to our research and is under review by the journal Wiley Interdisciplinary Reviews. Some studies seek to the analysis of most used apps in several contexts [Xu et al. 2011, Li et al. 2015]. However, such works do not aim to identify or monitor profiles and behaviors of users over time. Other studies have proposals for predic-

[1]Apps available in leading app stores - *goo.gl/3FLn4f*

ting users likely to churn (i.e turnover) [Rehman and Raza Ali 2014, Backiel et al. 2016]. Since the available data are in a batch fashion and the users are considered as independent individuals, most of these works consider the churn prediction task as a classification problem. Thus, usage profiles are not investigated as well as profile monitoring is not performed. These studies address different types of data, such as call data records (CDR), billing data and personal information (i.e. age and gender), not using app usage data. On the other hand, some studies propose the identification of usage profiles, which is carried out in several areas, such as Communication [Pyo et al. 2015] and Mobile [Rehman and Raza Ali 2014, Hamka et al. 2014]. Some of the studies from the mobile area do not use app usage data [Rehman and Raza Ali 2014] while others complement their datasets with this data [Hamka et al. 2014] but investigate only the number of apps used by users. Finally, in the last decade, some approaches were proposed for DS scenarios seeking to the identification and monitoring of clusters (i.e. profiles) on several areas [Spiliopoulou et al. 2006, Oliveira and Gama 2010]. However, such works are only intended to analyze changes between the clusters and not the behaviors of the objects (i.e users). We found a single study aiming to monitor usage profiles using a mobile DS [Pereira and Mendes-Moreira 2016]. However, such work does not use app data, applying only CDR.

3. Proposed Framework

In a real-world scenario, users perform a set of activities through a mobile device. Figure 1 (a) demonstrates the apps being run in the foreground by the user over time for three different devices. Each data event corresponds to an activity performed by a single user through the use of an app on a mobile device. Hence, a mobile DS contains million of activities or app events, which are produced from thousands of devices using one of the thousands of apps available in different time spans. A single activity, such as the app *WhatsApp*, is carried out repeatedly over time and such activities may be performed by several users and by different amounts of usage time (e.g. seconds or minutes). On the other hand, activities may or may not be carried out in the same time window (e.g., day or week) as shown in Figure 1 (b). According to the used time window, more or fewer events are processed and summarized. Thus, we can not consider each event as a complete representation of an independent object (device). It is how traditional data stream algorithms interpret them [Gama 2010].

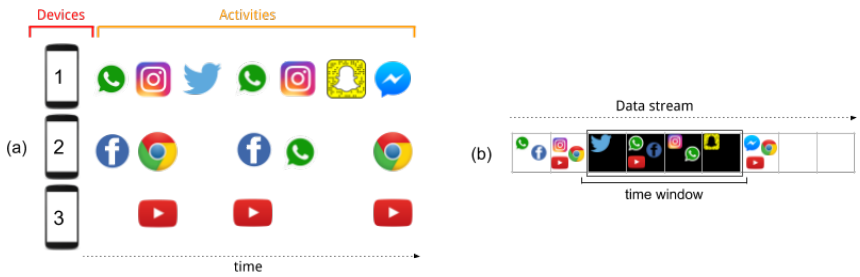


Figura 1. App Data Stream overview.

In this sense, the proposed framework aims to: collect app usage patterns, provide a limited number of usage profiles, and facilitate the monitoring of profiles and behaviors of users in a real-world scenario. To this end, our framework is composed by two main steps, namely *Data Stream Mining* and *Data Stream Monitoring* (see Figure 2).

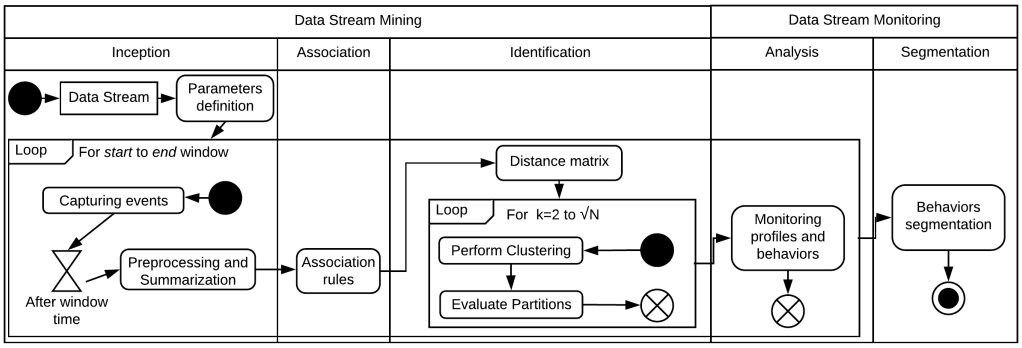


Figura 2. The Activity Diagram of the proposed framework.

3.1. Step 1: Data Stream mining

Inception phase - In the Inception phase, all events captured in the current time window are continuously preprocessed and summarised to handle with the several activities performed by each single user, as well as space and memory constraints. With a large number of app usage events, it is necessary to define a time window or a quantity of events which will be stored and preprocessed. This type of imposition is needed given the billion of events generated representing hundreds of terabytes stored in the physical memory. Time windows are the simplest way to maintain a practicable amount of data in physical memory. Such windows can help in the transition between data from the recent past and data from a distant past. Moreover, given a huge quantity of events, we require a data summarisation process to preserve the meaning of events without actually storing them [Gama 2010]. To this end, some parameters are defined in this phase: (i) the size of the time window that will cover the most recent data, (ii) a starting time, and (iii) an ending time for the data to be captured. For each window, we select the *most popular apps* and the *popular apps* based on the number of unique devices that use such apps. After, we perform the *Intuitive Partitioning* (IP) [Han et al. 2011] seeking to discretize the continuous values of the *popular apps*. The IP technique was more effective than others dividing such values into either a small or a large number of intervals. After the app usage mining, such data is summarized into a matrix composed of objects (devices) and attributes (apps) to be used as input for the Association rule algorithm in the next phase.

Association phase - We perform the Association Rules task [Tan et al. 2006] with the Apriori algorithm seeking to obtain positive correlations of app usage. We use these patterns to measure app usage similarity between the users aiming to use that in the identification of usage profiles. To the execution of this phase, it is necessary to receive as input the previously generated summary statistics as a transaction set. In addition, we need to define thresholds for support and all-confidence measures, which are used to generate itemsets, and for confidence and lift measures, which are used to generate the final rules.

Identification phase - Each itemset that composes one or more of the final rules has different users as support. For a user to be considered a support for a particular itemset, she should make use of all apps present on such itemset. In this sense, the distance between two users $u1$ and $u2$, who have their respective set of supported itemsets $i1$ and $i2$, is computed by the size of the intersection divided by the size of the union of the sample sets, as shown in Equation 1. In case of no existence of intersection between $i1$ and $i2$, the distance tends to infinity.

$$dist(u1, u2) = -log\left(\frac{|i1 \bigcap i2|}{|i1| + |i2| - |i1 \bigcap i2|}\right) \quad (1)$$

Such distance is computed for each pair of objects in a multidimensional space. Them, we obtain a matrix $y = [N \text{ x } N]$, where N is the number of observed users. In summary, the main goal of the first step is to make similar to each other all the users with the same usage patterns. Them, we carry out the Clustering task [Tan et al. 2006] aiming to identify usage profiles. The WARD algorithm receives as input a distance matrix between users based on the itemsets that generated the final rules in the previous phase. WARD produce several partitions, with a varying number of groups, and these partitions are evaluated using *Silhouette* and *Gap statistic*. We perform Data Mining and Unsupervised Machine Learning tasks aiming to extract the best patterns of the analyzed data regarding the huge amount of events produced by app usage. Thus, at the end of this step, all users have been mapped to one of the obtained clusters that are further investigated once they may change, evolve or disappear throughout time.

3.2. Step 2: Data Stream Monitoring

Analysis phase - In several real-world DS scenarios, it is necessary to monitor and investigate changes in the profiles (e.g. Concept Drift or Concept Evolution) and also of the behaviors of individuals composing these profiles. Indeed, it is necessary to distinguish or find same profiles in different time windows. It is feasible by tracking such concepts (i.e profiles) [Spiliopoulou et al. 2006, Oliveira and Gama 2010]. In this sense, we aim to perform Novelty Detection to discovery these variations that occur over the DS as well as to monitor users' behaviors given such changes.

In summary, we carry out Concept Drift and Concept Evolution approaches addressing the representation of each profile by the *enumeration* [Spiliopoulou et al. 2006] technique. Therefore, we monitor a profile by investigating the frequency distribution of its objects (users) in the next window profiles. On the other hand, it is necessary to understand the evolution of users through these profiles over time. In this sense, while each cluster label represents one profile on a single window, different labels may represent the same profile on several windows. Thus, in the next phase, we propose a new plan to analyze the changes in users' behaviors regarding the changes and evolutions of concepts found.

Segmentation phase - In this phase, we propose a sequence for the monitoring step aiming to track the changes in users' behaviors. We aim to find similar users' practices that allow segmenting such behaviors. The users are investigated to understand when and what their behavioral changes occur over the windows. In this sense, we define *life curves* seeking to represent the behavior of the users. For example, curves may show the continuity of a user in one profile, his change to other profile or the disappearance of such user. Therefore, here we aim to acquire knowledge to help us in the understanding of all users' behaviors. Moreover, we seek to segment users with same behavior and identify behaviors that could represent risk situations for mobile devices manufacturer companies.

Given a real scenario, a *life curve* is a temporal representation based on users that have the same behavior in the whole monitor process even in different profiles. To this end, some kinds of behaviors are designed (L, C, M, and O). The *L* action happens when

a user belongs to a profile c_{i+1} obtained in the window t_{i+1}, which represents the same profile c_i, to which this same user was grouped in the window t_i. The *C* action occurs when a user belongs to a profile c_{i+1}, obtained in the window t_{i+1}, which not represents the same profile c_i, to which this same user was grouped in the window t_i. The *M* action happens, when a user does not generate app usage events and such user is not grouped into the profiles obtained in the window t_{i+1}. And the *O* action occurs when a user is not supported by any itemsets obtained in the window t_{i+1}. Such behaviors are detected according to the evolution of the profiles helping us to analyze and understand how users behave throughout time.

4. Results and Discussion

We are improving our initial framework [Machado and Ruiz 2017] regarding to new experiments performed based on a real DS, as described next.

Step 1: We use a private DS provided by our sponsor, having 1,045,013,673 app events from 34,552 devices and 60,116 apps that were monitored for 140 days. A week time window (e.g. 7 days) was chosen for our experiments. Among all apps found in each week, it is possible to observe the existence of a substantial number of apps used by only a single or few devices. In this sense, we systematically explored how to define the *most used apps* for mining. To this end, we define the *most used apps* based on a minimal number of unique devices. Such apps are those used by 1% or more devices ($\overline{x}$ = 149). In addition, we define the *popular apps*, which are those used by 10% or more devices($\overline{x}$ = 33). *Popular apps* are widely used by users motivating their discretization. After the discretization process with IP approach the data are summarized into a matrix that is transformed in a transactinal set. At this step, we define thresholds for the above-mentioned association measures. Such definitions are based on the percentage of users used to define the *most used apps*. In this sense, the *support* is 0.01 and the *confidence* is 0.10. In addition, *all-confidence* and *lift* measures are both computed as the mean of all values found. This way, we obtain the patterns of app usage ($\overline{x}$ = 2,000), which are used accordingly to the Equation1 to compute the distance matrix. Finally, with the combination of *WARD*, *Silhouette* and *Gap statistic* we found the usage profiles for each window ($\overline{x}$ = 6).

Step 2: With the obtained result from step one, our framework begins to monitor the profiles and their objects aiming to detect changes in the learned concepts and in the users' behaviors. In order to distinguish potential risk situation, we design all possible *life curves*, based on the *life curve* of each user. Implementations of the approaches of this step have been improved based on additional experiments, with new results expected soon. In summary, we found 706 different *life curves* performing the *enumeration* approach. In this sense, the most frequent *curve* (e.g. more users) presents only *L* actions. Such *curve* indicates that most users have the same behavior and that behavioral changes are infrequent. Finally, we are performing experiments with new parameters to the *enumeration* approach seeking to improve the results for this step and evaluate our framework.

Discussion: Since the app usage activities are typically DS events, with changes in the data distribution, such data require accurate analysis, which does not occur in batch scenarios. In this scenario is crucial to use such DS-oriented analysis and also time win-

dows, allowing the understanding of customers characteristics that may be of interest to mobile device manufacturers and other stakeholders. For example, we have been able to identify customers profiles, variations in concepts and changes in users' behaviors. Thus, such knowledge becomes relevant and may be used strategically in the decision-making process by companies seeking to understand the behavior of their users. In this sense, such knowledge may indicate investment perspectives to keep the users loyal to their brands.

Acknowledgments: We gratefully acknowledge Motorola Mobility for its support to this research.

Referências

Almana, A. M., Aksoy, M. S., and Alzahrani, R. (2014). A survey on data mining techniques in customer churn analysis for telecom industry. *IJERA*.

Backiel, A., Baesens, B., and Claeskens, G. (2016). Predicting time-to-churn of prepaid mobile telephone customers using social network analysis. *JORS*, 67(9).

Gama, J. (2010). *Knowledge discovery from data streams*. CRC Press, Boca Raton.

Hamka, F., Bouwman, H., et al. (2014). Mobile customer segmentation based on smartphone measurement. *Telematics and Informatics*, 31(2):220–227.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33:1–26.

Li, H., Lu, X., et al. (2015). Characterizing smartphone usage patterns from millions of android users. In *ACM SIGCOMM*, pages 459–472.

Machado, N. L. and Ruiz, D. D. (2017). Customer: A novel customer churn prediction method based on mobile application usage. In *IEEE IWCMC)*, pages 2146–2151.

Oliveira, M. D. and Gama, J. (2010). Mec-monitoring clusters' transitions. In *STAIRS*, pages 212–224.

Pereira, G. and Mendes-Moreira, J. (2016). Monitoring clusters in the telecom industry. In *Springer WorldCIST*, pages 631–640. Springer.

Pyo, S., Kim, E., et al. (2015). Lda-based unified topic modeling for similar tv user grouping and tv program recommendation. *IEEE CYB*, pages 1476–1490.

Rehman, A. and Raza Ali, A. (2014). Customer churn prediction, segmentation and fraud detection in telecommunication industry. *ASE BD/SI/PASSAT/BMC Conf.*

Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y., and Schult, R. (2006). Monic: modeling and monitoring cluster transitions. In *ACM SigKDD*, pages 706–711.

Tan, P.-N., Steinbach, M., Kumar, V., et al. (2006). *Introduction to data mining*. Pearson Addison Wesley Boston.

Wagner, D. T., Rice, A., and Beresford, A. R. (2013). Device analyzer: Understanding smartphone usage. In *Springer MobiQuitous*, pages 195–208.

Xu, Q., Erman, J., et al. (2011). Identifying diverse usage behaviors of smartphone apps. In *ACM SIGCOMM*, pages 329–344.

An autonomous hybrid data partition for NewSQL DBs

Geomar André Schreiner[1]
Coorientador: Denio Duarte[2]
Orientador: Ronaldo dos Santos Mello[1]

[1]Programa de Pós-Graduação em Ciência da Computação
Departamento de Informática e Estatística– Universidade Federal de Santa Catarina
Florianópolis – SC – Brasil

[2]Universidade Federal da Fronteira Sul
Chapecó – SC – Brasil

schreiner.geomar@posgrad.ufsc.br, duarte@uffs.edu.br, r.mello@ufsc.br

Nível : Doutorado
Admissão : Março de 2016
Exame de Qualificação : Junho de 2018
Conclusão : Março de 2020
Etapas Concluídas : Revisão bibliográfica e projeto
Etapas Futuras : Implementação e avaliação

Abstract. *Several applications like online games and financial market seek support for features such huge data volumes management, data streaming and handle thousands of OLTP transactions per second. Traditional Relational Databases (RDBs) in general are not suitable for these requirements. NewSQL is a new generation of DBs that combines the high scalability and availability with the ACID support. NewSQL is a promising solution to handle these application requirements. Although data partition is an important feature for tunning relational DBs, it stills an open problem field for NewSQL systems. This Thesis proposes an automated approach for hybrid data partitioning that automatically reorganizes data based on the current workload of the NewSQL DBs.*

Resumo. *Várias aplicações, como jogos on-line e mercado financeiro, buscam suporte para recursos como gerenciamento de grandes volumes de dados, data streaming e suporte a milhares de transações OLTP por segundo. Bancos de dados relacionais tradicionais (RDBs), em geral, não são adequados para esses requisitos. BDs NewSQL são uma nova geração de bancos de dados que combina alta escalabilidade e disponibilidade com o suporte as propriedades ACID. Os NewSQL mostram-se uma solução promissora para lidar com esses requisitos de aplicações. Embora a partição de dados seja um recurso importante para melhoria no dempenho de sistemas NewSQL é ainda um problema aberto na literatura. Esta Tese propõe um controle da evolução do particionamento dos dados de um BD NewSQL com suporte a streaming de dados, de maneira autônoma, a fim de melhorar o desempenho de transações OLTP e de consultas de streaming.*

Palavras Chave: data partitioning, NewSQL, OLTP Systems, Automate partitioning, Big Data

1. Introduction

Historically, application systems rely on OLTP transactions to perform small operations over the network, such as buying an item in an online store [Stonebraker 2012]. The number of users that uses online stores has increasingly grown, and so the number of OLTP transactions performed. With the emergence of *Web*, OLTP requests has changed as well to deal with transactions on on-line games, social networks, or even large financial companies. These applications are characterized by having a large number of user interactions with the system, generating a huge amount of data and multiple OLTP transactions per second.

For decades, traditional Relational Databases (RDBs) have been used as an efficient way to store AND handle applications data, but they are not suitable to handle these massive data volume associated with high availability and ACID support guarantees [Stonebraker 2012]. Based on problems faced when Big Data needs to be managed new architectures are emerging like NoSQL DBs. These new architectures were born in cloud environments, generally, capable of storing and managing huge data volumes, keeping high availability and scalability. NoSQL solves part of the problems related to Big Data management, they maximize availability rather than ACID support. Companies keep using RDBs for most of their data-applications because it is simpler to deal with the overhead of traditional ACID assurance than with the lack of these properties. With the goal of offering availability, scalability and ACID support, the *NewSQL* movement has arisen.

NewSQL DBs main advantage is try to combine the best of both worlds: scalability and availability of NoSQL DBs with ACID properties of traditional RDBs [Stonebraker 2012]. Usually, NewSQL DBs are distributed in-Memory DBs, and each node of the DB has a partition of the stored data [Pavlo and Aslett 2016, Taft et al. 2014, Elmore et al. 2015, Kallman et al. 2008]. Data partitioning in distributed systems affects directly the system performance, since data need to be reconstructed or merged to attend DBs operations. Thus, the proposition of efficient approaches to control data partitioning, in order to maximize access performance and OLTP processing, becomes essential in this type of data management system.

Generally, relational data can be partitioned in two ways: horizontally (by rows) and vertically (by columns). Both types of partition vertical and horizontal can improve query performance in different ways [Al-Kateb et al. 2016]. For example, when a selection is performed, if the tuples are in the same partition,the network traffic is optimized. The same reasoning may be used to the projection operation, vertical partition optimizes data access as well. We found some initiatives in literature that explore the use of vertical partition in OLTP transactions [Amossen 2010] and hybrid partitioning in systems that consider OLAP transactions [Arulraj et al. 2016], but data partitioning stills an open field for NewSQL systems.

H-Store [Kallman et al. 2008] and *S-Store* [Cetintemel et al. 2014] are NewSQL data management systems that consider a pre-workload plan to define an optimized partitioning. However, the data volume and workload of a DB are not static (*i.e.*, constantly change) and these proposals do not consider the evaluation and modification of these partitioning strategies. Other proposals, such as *Clay* [Serafini et al. 2016], *Accordion* [Serafini et al. 2014] and *E-Store* [Taft et al. 2014] have repartitioning tools. All

three have an autonomous system that just considers OLTP loads. However, none of them has considered an hybrid data partition or streaming application requirements.

In this Thesis, we propose a new automated partitioning system that evolves the partition scheme of a NewSQL DB. We consider hybrid data partition for minimizing distributed transactions on OLTP systems and considering data streaming requirements. Our proposal is a reactive system capable of generating new data partition schemes based on the workload of the system and reorganize data with no down time seeking for a best performance of the NewSQL system. Expected contributions of our work are: (*i*) a novel approach for data partitioning that considers hybrid data partition for OLTP and streaming operations minimizing the distributed transactions; (*ii*) partitioning based in a fine-grained level, storing tuples in an hybrid scheme (vertical and horizontal) based workload access; (*iii*) an adaptive approach that allows the system to reorganize data based on the current workload; and (iv) an approach that also considers replicas to increase availability and therefore decreases the number of distributed transactions.

The rest of this paper is organized as follows. First we present our propose in a high level structure showing a generic NewSQL architecture and our propose partition algorithm. Section 3 discuss related works. Section 4 presents the current status of our work and some future activities.

2. Proposal

Figure 1 presents the general architecture of the NewSQL system, and it is basically a distributed architecture with two components: *(i)* a *master* node (dotted border), and *(ii) worker* nodes named *sites* where the data partitions are stored.

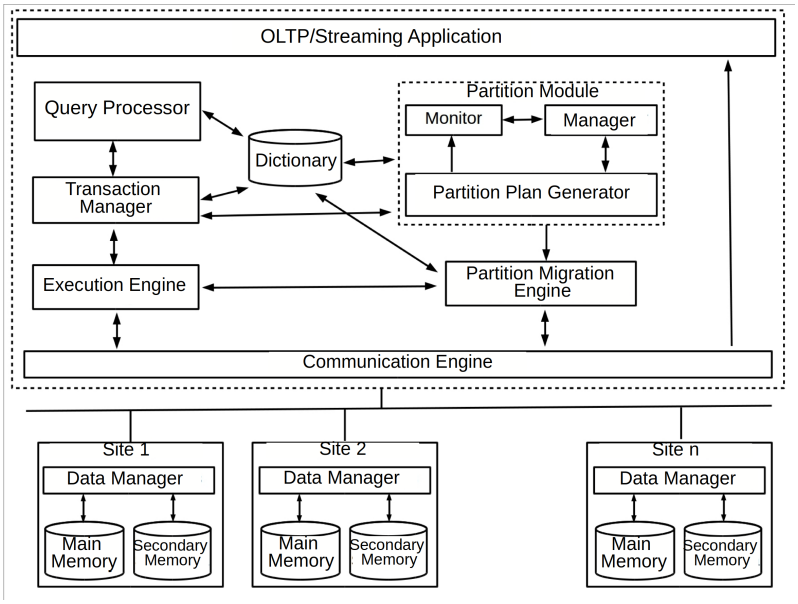


Figure 1. Proposed architecture

The *master* node receives all requests sent by applications through a specific *interface* (OLTP / Streaming Application module - Figure 1). The application connects to this interface and sends its requests and receive the responses. The master node is composed of: (*i*) Data dictionary that stores meta-data information such as data schema,

which partitions the data is located, how often it is used, *stored procedures*, and updated information about the *workload*; (*ii*) Query Processor that receives a given SQL command and defines a query plan using meta-data information stored in the Dictionary; (*iii*) Transaction Manager which ensures the efficient execution of the query plan, or a transaction in general, in the correct order; (*iv*) Partitioning Module, part of our contribution, for generate new optimized partitioning scheme; (*v*) Execution Engine that is responsible for executing each of the operations requested by the Transaction Manager and to control the *streaming*-based queries; (*vi*) Partition Migration Mechanism which receives a new partition plan to migrate the data accordingly; and (*vii*) Communication Engine module which performs the communication between the master node and worker nodes in order to execute the operations that the Execution Engine requests.

In general, NewSQL systems aim to handle single node transactions by avoiding distributed transactions, so the *Transaction Manager* tries to assure the execution order of transactions in a single *site* otherwise, it coordinates the distribution of the operations among the *sites* and assures the ACID properties.

The Sites (Site 1 to n in Figure 1) have a simple structure and are used to store the data partitions, one partition per site. Sites are composed of only one module *Data Manager* that uses *Main Memory* (volatile and fast access) and *Secondary Memory* with slower access) to manage the data. *Data Manager* is responsible for receiving demands coming from the *master* node and managing which data is in main memory and which data should remain on the disk.

2.1. Partition Module

The Partitioning Module is responsible for monitoring the performance of the approach and creating optimized partitioning plans for the application demands. The module is organized into three components: *Monitor*, *Manager* and *Partition Plan Generator* (see Figure 1).

The *Monitor* component collects information about the overall state of the system in real time (is always active by collecting information about the overall state of the system). It collects access statistics information about the data schema present in the *Dictionary* as well as other information about the types of transactions executed directly in the *Transaction Manager* module. The information collected serves as the basis for *Manager* component decision and is used by the *Partition Plan Generator* component to optimize data partitioning.

Based on the information collected by *Monitor*, the *Manager* component decides whether or not a partition migration is required. The *Manager* analyses workload of the system using some heuristics to identify (possible) unbalancing points or critical points (recurrent distributed transactions) and calls the *Partition Plan Generator* for a new partition plan. With a new partition plan, the *Manager* checks, through a cost function, if the data migration does not present a very high cost for the performance gain. If the migration cost is too high and the performance gain is too small, the system does not migrate the partition.

We use a heat graph structure to help the *Partition Plan Generator* to create new partition plans. The heat graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph partitioned in n partitions composed of a set of vertices $\mathcal{V}$ and a set o edges $\mathcal{E}$. Each vertice $\nu \in \mathcal{V}$ represents an accessed

tuple. ν is composed of a tuple $< n_\nu, n_{table}, Attrs, k >$ where n_ν is the identifier of tuple (*rowid*), n_{table} name of the table that the tuple belongs, $Attrs$ is the set of attributes accessed by the tuple (a projection of SQL statement), and k is the number of times that the tuple was accessed, the value of k indicates the tuple temperature (bigger the value hottest is the tuple). Each edge $\epsilon \in \mathcal{E}$ represents tuples that were accessed together in a transaction, and ϵ is composed of a tuple $< \nu, w, \nu' >$ where ν and ν' are two vertices that represents two tuples accessed together, and w number of times that the tuples were accessed (weight). The same tuple can be accessed several times with different projections, for each different projection, a new vertice is created with the tuple identifier and the access projection. Vertices that represent the same tuple (have the same *identifier*) are connected by edges with no weight (w=0). The graph $\mathcal{G}$ is partitioned in n partition, each partition maps one *site* of the NewSQL systems.

To generate a new partition plan the *Partition Plan Generator* first analyzes $\mathcal{G}$ to search for unbalanced partitions. As described previously, each partition of $\mathcal{G}$ maps the *sites* and respectively tuples (*vertices*). To detect unbalancing partitions we use a function that sums all vertices weights for each partition, if a partition have a total weight very different from the others, then it is unbalanced. When an unbalancing partition is detected, the approach selects the overloaded partition and analyzes the hottest tuples, searching for one tuple t that can migrate to an underloaded partition. Based on $\mathcal{G}$ a new graph $\mathcal{G}'$ is created migrating t to the new partition. Each edge that connects t to other tuples t' in a different partition is considered as a critical edge. We analyze all critical edges (edges that connect vertices from different partitions) and reallocate them to the same partition. That is, tuples that are accessed together but are in different partitions should be in the same partition. We try to keep as few as possible vertices belonging to different partitions to be connected by heavy edges, *i.e.*, giving $<\nu, w, \nu'>$ and $\nu \in p_1$ and $\nu' \in p_2$ ($p_1 \neq p_2$), the closer w is to 0, the better. Finished this phase, $\mathcal{G}'$ is translated by a decision three in a partition plan and sent to *Manager* component. The *Manager* component, using a cost function, analyzes if $\mathcal{G}'$ partition plan is better than $\mathcal{G}$. If $\mathcal{G}'$ is better than $\mathcal{G}$, then data migrating tool is called to reorganize the partitions.

If no unbalancing point is detected the approach analyses the critical edges of $\mathcal{G}$. If a critical edge e is found with a high weight, the approach evaluates the migration options for the vertices connected by e. A new graph $\mathcal{G}'$ is created based on $\mathcal{G}$, and $\mathcal{G}'$ is used to migrate tuples among partitions.

Our migration approach will be based on the Squall [Elmore et al. 2015]. Squall makes a fine-grained migration, at tuple level with no downtime. Squall synchronizes all sites and each *site* is responsible for evaluating the partition plan and the routing table (that stores where each tuple is) to migrate their data The migration transactions are routed to the target *site* and added with no priority on its transactions queue. Eventually, if a transaction T is routed to a *site*, and the data is waiting for migration, then T is put in a lock state, the site execute the migration immediately , and then T is unlocked and execute their operations.

3. Related work

Our approach fits into related work that offers a way of data partition for systems with the main focus in OLTP transactions (NewSQL DBs). All ap-

proaches found in the literature that propose partition approaches for OLTP-based systems use horizontal partitioning to organize the data, that is, they all split the tables into sets of tuples, and each partition maintains a subset of the table. *H-Store* [Kallman et al. 2008], *Horticulture* [Pavlo et al. 2012], *Accordeon* [Serafini et al. 2014], and *S-Store* [Meehan et al. 2015] partition their tuples horizontally using the identifier of each tuple. Each partition stores a set of tuples with sequential identifiers. On the other hand, *Schism* [Curino et al. 2010], *E-Store* [Taft et al. 2014], and *Clay* [Serafini et al. 2016] group tuples by affinity. In this way, tuples usually accessed together will be stored in the same partition.

Most of the approaches are based on autonomous partitioning. *H-Store* and *S-Store* leave it to the DBA to create or use an external tool that will generate suitable partitioning for the desired demands. However, other approaches have support to the autonomous partitioning aiming at decreasing the distributed transactions. *Horticulture* only takes into account the operations provided by the DBA and the data schema to feed its LNS (Large-Neighborhood Search) algorithm, which generates the partitioning. *E-Store* partitions its data using some statistics to identifying the most accessed tuples. These tuples are allocated in different nodes, balancing the loads. *Accordion*, in addition to statistics generated from the workload, takes into account the maximum capacity of each server and groups the partitions accessed together on the same server or nearby servers. Unlike the others, *Clay* and *Schism* use a heat graph to accomplish this task. This graph is created based on the tuple access. Each tuple is represented by a vertice with a temperature (number of times the tuple was accessed). Edges connecting vertices represent tuples accessed together in the same transaction.

Only three of the approaches use replicas of data to improve partitioning performance. *Horticulture* and *Schism* create a partitioning plan that takes into account block replication to facilitate transaction execution and increase system availability. *S-Store*, although it does not have a repartitioning model, performs data replication for *streaming* support. Data accessed by a *streaming* operation are replicated into tables created exclusively for this purpose.

Based on the literature review, we can see the lack of an approach that explores some specific aspects: *(i)* a hybrid tuples partitioning that decides when it is advantageous to use vertical or horizontal partitioning; *(ii)* a solution that encompasses data streaming and OLTP load issues, enabling the system to meet the requirements of a wide range of applications; *(iii)* a solution that considers replicas to increase availability and therefore decreases the number of distributed transactions; and *(iv)* a solution that performs a gradual and periodic evolution of the existing partitions, avoiding periods of high latency until its partitioning is optimized.

4. Final Considerations

This thesis proposes an autonomous approach for data partitioning for NewSQL BDs, taking into consideration the *workload*, OLTP loads and data streaming support. The proposed approach is unprecedented in the literature for generating a hybrid data partition scheme (vertical and horizontal), offering data optimization and data storing based on access workload, furthermore we propose a partitioning algorithm that considers data replication to reduce overhead of distributed transactions.

This Thesis is in half way of its development, so we have some future works in mind. In current status, we are developing support for vertical/hybrid partitioning in *VoltDB*. The next steps involve the partition module development for VoltDB to validate the ideas of the approach. To validate our approach, we will use consolidated benchmarks (TPC-W, YCSB) comparing with state-of-the-art approaches. We also plan to evaluate the use of machine learning techniques to predict when a partition reorganization should be triggered.

References

Al-Kateb, M., Sinclair, P., Au, G., and Ballinger, C. (2016). Hybrid row-column partitioning in teradata®. *Proc. VLDB Endow.*, 9(13):1353–1364.

Amossen, R. R. (2010). Vertical partitioning of relational oltp databases using integer programming. In *ICDEW*, pages 93–98. IEEE.

Arulraj, J., Pavlo, A., and Menon, P. (2016). Bridging the archipelago between row-stores and column-stores for hybrid workloads. In *SIGMOD 2016*, New York, NY, USA. ACM.

Cetintemel, U., Du, J., Kraska, T., and Madden, e. a. (2014). S-store: A streaming newsql system for big velocity applications. *Proc. VLDB Endow.*, 7(13).

Curino, C., Jones, E., Zhang, Y., and Madden, S. (2010). Schism: A workload-driven approach to database replication and partitioning. *Proc. VLDB Endow.*, 3(1-2).

Elmore, A. J., Arora, V., Taft, R., Pavlo, A., Agrawal, D., and El Abbadi, A. (2015). Squall: Fine-grained live reconfiguration for partitioned main memory databases. In *2015 ACM SIGMOD*, pages 299–313, New York, NY, USA. ACM.

Kallman, R., Kimura, H., Natkins, Stonebraker, M., et al. (2008). H-store: a high-performance, distributed main memory transaction processing system. *VLDB*, 1(2).

Meehan, J., Tatbul, N., Zdonik, S., Aslantas, C., et al. (2015). S-store: Streaming meets transaction processing. *Proc. VLDB Endow.*, 8(13).

Pavlo, A. and Aslett, M. (2016). What's really new with newsql? *SIGMOD Rec.*, 45(2):45–55.

Pavlo, A., Curino, C., and Zdonik, S. (2012). Skew-aware automatic database partitioning in shared-nothing, parallel oltp systems. In *2012 ACM SIGMOD*, pages 61–72, New York, NY, USA. ACM.

Serafini, M., Mansour, E., Aboulnaga, A., Salem, K., Rafiq, T., and Minhas, U. F. (2014). Accordion: Elastic scalability for database systems supporting distributed transactions. *Proc. VLDB Endow.*, 7(12):1035–1046.

Serafini, M., Taft, R., Elmore, A. J., Pavlo, A., Aboulnaga, A., and Stonebraker, M. (2016). Clay: Fine-grained adaptive partitioning for general database schemas. *Proc. VLDB Endow.*, 10(4):445–456.

Stonebraker, M. (2012). New opportunities for new sql. *Commun. ACM*, 55(11).

Taft, R., Mansour, E., Serafini, M., Duggan, J., Elmore, A. J., Aboulnaga, A., Pavlo, A., and Stonebraker, M. (2014). E-store: Fine-grained elastic partitioning for distributed transaction processing systems. *Proc. VLDB Endow.*, 8(3).

Avaliação da Saúde de Ecossistemas de Dados

Aluna: Glória de Fátima Andrade Barros Lima
E-mail: gfabl@cin.ufpe.br

Orientador: Bernadette Farias Lóscio
E-mail: bfl@cin.ufpe.br

Co-orientador: Marcelo Iury S. Oliveira
E-mail: miso@cin.ufpe.br

Nível: Mestrado

Universidade Federal de Pernambuco - UFPE

Programa de Pós-graduação em Ciência da Computação – Centro de Informática

Mês e Ano de ingresso: Março de 2017

Mês e Ano previstos para defesa: Março 2019

Etapas Concluídas: Créditos em disciplinas, Definição do tema e Levantamento do Referencial Bibliográfico

Etapas Futuras: Finalizar o levantamento e especificação dos critérios e métricas para avaliação da saúde de Ecossistemas de Dados, Formalizar a estratégia de avaliação de saúde em Ecossistemas de Dados, Avaliação da estratégia proposta.

Resumo. *A disponibilização de dados em meio digital, por entidades públicas e privadas, tem crescido bastante nos últimos anos, contribuindo para a geração de valor por meio do compartilhamento e consumo desses dados. Nesse contexto, surgem os Ecossistemas de Dados (ED), que podem ser definidos como redes de atores autônomos que consomem, produzem ou fornecem direta ou indiretamente dados e outros recursos relacionados aos dados (e.g., software, serviços e infraestrutura). Apesar do grande número de Ecossistemas de Dados atualmente disponíveis, nem todos podem ser considerados saudáveis e sustentáveis, dificultando o compartilhamento e a troca de recursos pelos atores do ecossistema. Dessa forma, avaliar a saúde desses ecossistemas torna-se fundamental para prevenir problemas no seu funcionamento, bem como para garantir a sua sustentabilidade ao longo do tempo. A saúde de um ED pode ser avaliada por meio de critérios, como: sustentabilidade, resiliência e geração de valor. Na literatura não foram encontrados trabalhos com este propósito. Neste trabalho, propomos uma estratégia para avaliação da saúde de Ecossistema de Dados consistindo de um conjunto de critérios e métricas que permitem uma avaliação contínua da saúde do ecossistema. Com o uso da estratégia proposta, espera-se obter um conjunto de indicadores para auxiliar nas tomadas de decisão relacionadas aos processos de publicação e consumo de dados, bem como avaliar quais ecossistemas são mais prósperos ou carecem de maiores investimentos.*

1. Introdução e Motivação

Nos últimos anos, estudos voltados para Ecossistemas de Dados (ED) vêm crescendo por causa da relevância que os dados apresentam nas tomadas de decisão em diferentes contextos, desde ambientes governamentais até ambientes corporativos. A disponibilização de dados por governos e entidades privadas tem crescido e facilitado o consumo, a troca e a produção de recursos que utilizam esses dados como meio para gerar valor e alavancar o crescimento. Um ED pode ser visto como "uma rede de atores autônomos que diretamente ou indiretamente consomem, produzem ou fornecem dados e outros recursos relacionados a dados (e.g. software, serviços e infraestrutura). Cada ator desempenha um ou mais papéis e está conectado a outros atores por meio de relacionamentos, de forma que a colaboração e competição entre os atores promove a auto-regulação do Ecossistema de Dados" [Oliveira and Lóscio 2018].

Como a dimensão e a complexidade dos EDs são variáveis, a identificação de aspectos que impactam negativamente nesses ecossistemas se torna uma tarefa desafiadora. Sendo assim, como forma de identificar deficiências no funcionamento dos EDs, faz-se necessário ter indicadores que reflitam a saúde dos EDs. Neste trabalho, definimos como saúde do ED a sua capacidade de persistir e permanecer produtivo ao longo do tempo, tolerar a dinamicidade e variabilidade dos componentes e suprir as necessidades dos atores envolvidos.

Por se tratar de um tema recente, não existem trabalhos que avaliem especificamente a saúde dos EDs [Oliveira et al. 2017]. De maneira geral, a avaliação da saúde de ecossistemas é citada na literatura como uma forma de produzir indicadores operacionais sobre as atividades do ecossistema, os status dos elementos que o compõem, bem como gerar diretrizes sobre a governança do ecossistema [Oliveira et al. 2017] [Alves et al. 2017]. Entretanto, mais uma vez ressaltamos que a formalização de métricas específicas para avaliar a saúde dos EDs é um tópico ainda não explorado.

Nesse contexto, este trabalho propõe uma estratégia para avaliação da saúde de EDs, consistindo de um conjunto de dimensões, critérios e métricas que permitem avaliar o ED e, como consequência, geram um conjunto de indicadores sobre a saúde do ecossistema. Tendo em vista a ausência de trabalhos nesta área, usamos como inspiração propostas para avaliação da saúde de ecossistemas de negócios, ecossistemas de software e ecossistemas naturais.

Este trabalho está organizado como segue: a Seção 2 introduz os trabalhos relacionados, a Seção 3 apresenta nossa proposta, a Seção 4 detalha o método de pesquisa e a Seção 5 discute os próximos passos para a conclusão do trabalho.

2. Trabalhos Relacionados

[Costanza 1992] define que um ecossistema natural é saudável se ele for estável e sustentável; mantendo sua organização e autonomia ao longo do tempo e sua resiliência ao estresse. [Schaeffer et al. 1988] caracterizam um ecossistema em termos de estrutura e função, na qual a estrutura é identificada como os componentes físicos do ecossistema e a função como as atividades desempenhadas no ecossistema, por exemplo. A estrutura e a função são as dimensões consideradas nos ecossistemas naturais para avaliar qual o impacto que uma exerce sobre a outra.

Na literatura de ecossistemas de negócios, o conceito de saúde é definido como a habilidade desses ecossistemas proverem oportunidades de crescimento duráveis para seus membros e para aqueles que dependem dele [Iansiti and Levien 2004a]. [Iansiti and Levien 2004b] definem três escalas, conhecidas como PRN, para representar a saúde de ecossistemas de negócios: a produtividade, a robustez e a criação de nicho ou inovação, as quais são avaliadas por meio de métricas que ponderam a habilidade de converter materiais brutos em novos produtos e de menor custo, de ser resiliente e de aumentar a diversidade de atores ao longo do tempo. Já [den Hartigh et al. 2006], inspirados pelos trabalhos iniciais da área, definem a saúde de ecossistemas de negócios como o bem-estar financeiro e a força da rede de conexões, e dividem a saúde em dois componentes: a saúde do parceiro e a saúde da rede. A saúde do parceiro avalia a saúde de cada ator individualmente e a saúde da rede de conexões é medida pela conectividade dos atores.

Na literatura de ecossistemas de software, a principal fonte de inspiração são os ecossistemas de negócios, pois ambos possuem parâmetros similares de avaliação de saúde. [Manikas and Hansen 2013] estabelece um *framework* para avaliação da saúde de ecossistemas de software definindo os atores, o software e o *orchestration* como os três principais componentes que afetam a saúde desses ecossistemas, eles consideram a saúde dos componentes de forma individual e coletiva. De maneira mais específica, [Wahyudin et al. 2007] definem três critérios que influenciam a saúde de projetos *open source*, que são a vivacidade da comunidade de desenvolvedores, a vivacidade da comunidade de usuários e a qualidade do produto, no qual cada critério é avaliado por métricas específicas.

Entretanto, nenhuma das soluções apresentadas se adéqua totalmente à realidade dos EDs por causa dos componentes e das relações existentes nesses ecossistemas. Por outro lado, o crescimento da quantidade de dados disponibilizados e consumidos direta ou indiretamente por atores diversos está alavancando a produção de novos recursos e revelando a necessidade de se ter ambientes saudáveis para lidar com essa nova dinâmica.

3. Ecossistemas de Dados

Como mencionado anteriormente, os EDs podem ser definidos como redes de atores autônomos que consomem, produzem ou fornecem direta ou indiretamente dados e outros recursos relacionados aos dados (e.g., software, serviços e infraestrutura). No trabalho de [Oliveira and Lóscio 2018] foram identificados os principais componentes de um ED, os quais estão descritos a seguir. Estes elementos são a base para a escolha dos critérios e das métricas mais adequados para avaliação da saúde de um ED.

Atores: são entidades autônomas que exercem um ou mais papéis e são considerados elementos básicos do ED. Um conjunto de interesses motiva os atores e cada um tem expectativas diferentes. Os atores geralmente se comprometem com o ecossistema e precisam de incentivos para permanecer ativos no ecossistema.

Recursos: são os produtos, possessões ou capacidades úteis ou de valor, que são produzidos, providos, curados ou consumidos pelos atores. Os recursos podem ser trocados individualmente ou em conjunto por meio de transações.

Papéis: são funções exercidas por um ator no ED. Estão relacionadas com um

conjunto de deveres e atividades. Diversos papéis podem ser identificados nos ED, como consumidores, produtores e intermediários.

Relacionamentos: são as interações entre os atores, entre os recursos e entre os atores e os recursos dos EDs. Relacionamentos são geralmente baseados em interesses em comum ou são relacionados aos papéis que cada ator exerce no ecossistema.

Como cenário motivacional para exemplificar os componentes de um ED, suponha um ecossistema de dados de pesquisa acadêmico. Nesse cenário, os dados do contexto acadêmico são os protagonistas do ecossistema pois auxiliam os atores a produzir, trocar e consumir recursos. Os recursos oriundos de dados de pesquisa podem ser os próprios conjuntos de dados, os trabalhos acadêmicos ou os serviços e sistemas baseados nesses dados. Nesse ecossistema existem atores que lidam direta ou indiretamente com os recursos, alguns exemplos são os pesquisadores, entidades de pesquisa, financiadores, cidadãos, onde cada ator exerce um ou mais papéis, como os papéis de consumidor, produtor ou intermediário. Com isso, os relacionamentos entre os atores podem acontecer quando há interesses em comum (*e.g.* procura por um conjunto de dados específico) ou inerentes aos papéis (*e.g.* produção de soluções).

4. Solução Proposta

Este trabalho propõe uma estratégia para Avaliação da Saúde de Ecossistemas de Dados, a qual objetiva a produção de indicadores que podem ser utilizados para avaliar aspectos qualitativos e quantitativos relacionados à produtividade, resiliência e geração de valor de um ED, por exemplo. A produção de indicadores é alcançada com a aplicação de métricas para medir aspectos e elementos em EDs reais.

Apesar do grande número de EDs atualmente disponíveis, nem todos são ecossistemas saudáveis e sustentáveis, dificultando assim o compartilhamento e a troca de dados pelos atores do ecossistema. Por isso, ter uma estratégia para avaliar a saúde dos EDs e identificar quais aspectos não estão com um bom desempenho é essencial para que o ecossistema consiga sobreviver e ser produtivo por mais tempo.

Para a definição de quais componentes representam os EDs nos baseamos no trabalho de [Oliveira and Lóscio 2018]. Complementando os componentes principais apresentados, nós consideramos que os componentes ator e recurso, precisavam ser divididos em componentes menores, pois possuem peculiaridades que afetam a saúde do ED de formas diferentes. A saúde dos atores pode ser dividida em saúde individual do ator e saúde da rede de atores, e similarmente, a saúde dos recursos pode ser dividida em saúde dos conjuntos de dados, saúde das soluções baseadas em dados e saúde da rede de recursos. Com base nos componentes identificados e no escopo de avaliação de saúde, nós decidimos tratar os componentes como dimensões, para que posteriormente pudéssemos atribuir critérios e métricas para avaliar essas dimensões. Sendo assim, na Figura 1 propomos o detalhamento das principais dimensões que influenciam na saúde geral dos EDs e explicamos a seguir cada uma.

Saúde dos Atores Individuais: Esta dimensão avalia como, em ecossistemas de negócios, a produtividade, a robustez e a geração de valor dos atores, ainda que individualmente, influenciam a saúde do ED. Os atores possuem características que podem ser medidas e assim ter a saúde avaliada, como: os atributos que os caracterizam, suas expectativas em relação ao ED, suas capacidades, os papéis que eles exercem e os recursos

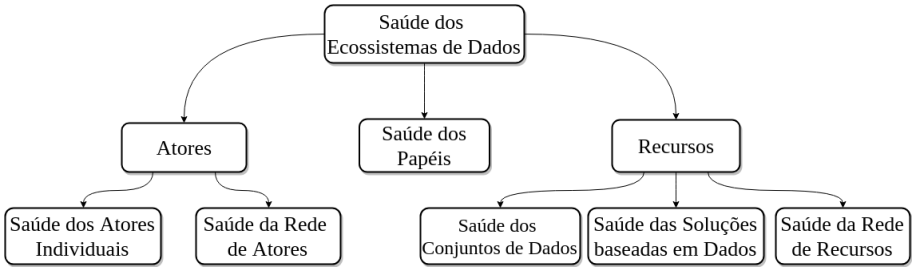


Figura 1. Dimensões para Avaliação de Saúde dos EDs

produzidos, providos ou consumidos por eles. Na Figura 2 exemplificamos os critérios e aspectos que podem ser considerados no cálculo das métricas para avaliar a saúde da dimensão de atores individuais.

Saúde da Rede de Atores: A rede de atores e suas interações têm um papel importante na saúde dos EDs. Uma rede de atores compreende um conjunto de atores individuais que se relacionam, exercem papéis diferentes, seguem modelos de negócio para entregar valor e podem ter um ou mais atores-chave que coordenam suas transações. As interações existentes variam de acordo com o contexto (e.g. econômico, político, cultural e tecnológico), entretanto, elas possuem algumas características que podem ser avaliadas para definir a saúde da rede de atores, como os tipos de transações existentes, modelos de negócios, se há atores que coordenam a rede e qual o impacto disso.

	Critérios	Aspectos a serem Considerados
Atores Individuais	Produtividade	Ativo em quantos projetos; Transações que participa; Interações com outros EDs; Contexto inserido; Papéis que desempenha;
	Resiliência	Quanto tempo participa do ED; Dificuldades ao desempenhar seus papéis; Conflitos superados;
	Geração de Valor	Participação na entrega de recursos; Exerce papel de liderança; Utilização de modelos de negócio; Variedade nos projetos;

Figura 2. Critérios para Avaliar a Saúde de Atores Individuais

Saúde dos Conjuntos de Dados: Os conjuntos de dados são o agrupamento de dados relacionados ou não que têm a função principal de ser fonte de informação. A saúde dos conjuntos de dados pode ser avaliada utilizando métricas inspiradas nas melhores práticas dos dados na Web (*Data on the Web Best Practices*) [Lóscio et al. 2017], como a qualidade dos dados, formatos disponíveis e facilidade de acesso.

Saúde das Soluções baseadas em Dados: As soluções baseadas em dados são os recursos produzidos a partir do consumo dos conjuntos de dados, sejam elas aplicações, serviços, entre outros. A saúde das soluções baseadas em dados abrangem as métricas de saúde dos conjuntos de dados e também métricas que avaliam o processo de criação de ideias e desenvolvimento, entrega de valor e suas dificuldades.

Saúde da Rede de Recursos: Uma rede de recursos representa as relações de dependência entre esses recursos e a influência que eles exercem nas outras dimensões dos EDs. Por exemplo, aplicações baseadas nos conjuntos de dados produzidos pelo mesmo ED, ou um conjunto de serviços que faz uso de uma infraestrutura desenvolvida por outro ED. A saúde da rede de recursos pode ser avaliada de acordo com diversas métricas, por exemplo pelo nível de dependência entre os recursos, como eles podem impactar no pro-

cesso de criação de novas soluções, quais recursos são considerados prioritários naquele ED e como essa rede de recursos influencia nas decisões do ED.

Saúde dos Papéis: Os papéis são as funções que os atores estão exercendo naquele momento, podendo um ator exercer mais de um papel no mesmo ED ou em outros. Existem diversos papéis nos EDs, geralmente os mais relatados na literatura de ED são os consumidores e publicadores de dados. Para definir a saúde de cada papel é preciso definir os papéis existentes no ED em questão, avaliar as atividades, se há hierarquias nas tomadas de decisão, como são atribuídos os papéis e a influência que cada papel exerce sobre o outro.

Cada dimensão apresentada representa um aspecto válido que contribui para o funcionamento e gerenciamento do ecossistema. Após a formalização dos critérios de avaliação será feito o estudo de como cada critério será calculado e consequentemente realizar a operacionalização dos cálculos. Com isso serão obtidos indicadores que revelem o estado da saúde do ED e assim ajudar a identificar quais componentes ou atividades podem estar prejudicando o sucesso do ecossistema. E finalmente, para validar a estratégia proposta, serão realizados dois estudos de caso em EDs reais.

5. Método de Pesquisa

Este trabalho visa investigar e desenvolver uma estratégia de medição e avaliação da saúde de EDs. Nosso *design* de pesquisa pode ser definido como um paradigma de pesquisa pragmático que visa construir artefatos originais e inovadores para resolver problemas do mundo-real [Von Alan et al. 2004]. Isso envolve a criação e a avaliação de artefatos para resolver um problema específico. Baseados nos nossos objetivos e no paradigma pragmático, nós dividimos o design de pesquisa em três fases:

- A primeira fase visa mapear elementos e aspectos qualitativos e quantitativos citados na literatura que influenciam a saúde de EDs. Esses elementos e aspectos serão usados na construção de métricas que podem determinar a saúde de um ecossistema de dados.
- A segunda fase da pesquisa visa a formalização da estratégia de avaliação da saúde em EDs. Esta fase da pesquisa irá formalizar o problema a ser resolvido, definindo formalmente as métricas de avaliação, assim como a operacionalização do cálculo das métricas.
- A terceira fase da pesquisa consiste na validação da estratégia proposta. Serão realizados dois estudos de caso nos quais a estratégia será aplicada em dois EDs reais (ED da Universidade Federal de Pernambuco e o ED da cidade do Recife-PE) para avaliar a saúde de seus respectivos ecossistemas. E, posteriormente, esses participantes irão responder questionários de avaliação da estratégia com base em um conjunto de critérios de qualidade.

6. Trabalhos Futuros

Neste trabalho a necessidade de criar uma estratégia para avaliar a saúde de Ecossistemas de Dados foi apresentada. Enquanto trabalhos na área de ED salientam a necessidade de avaliar a saúde desses ecossistemas [Oliveira et al. 2017], nenhum trabalho com este propósito foi encontrado. As estratégias de avaliação da saúde de ecossistemas encontrados na literatura não são capazes de lidar com a heterogeneidade e dinamicidade dos componentes (atores, relacionamentos, recursos e papéis) presentes nos EDs.

Como próximos passos neste trabalho nós pretendemos: (i) finalizar o levantamento e especificação dos critérios e métricas para avaliação da saúde de EDs; (ii) formalizar a estratégia de avaliação de saúde em ecossistemas de dados; (iii) validar a estratégia proposta.

Como parte da pesquisa sobre Ecossistemas de Dados foi publicado o artigo "A Platform for Supporting Open Data Ecosystems"no ICEIS 2016. O plano para futuras publicações inclui: (i) "Investigations about data ecosystems: A systematic mapping study"submetido para o KAIS e (ii) "Investigations about Data on the Web Publication and Consumption: A Systematic Mapping Study" para o WWW journal.

Referências

Alves, C., Oliveira, J., and Jansen, S. (2017). Software ecosystems governance-a systematic literature review and research agenda. vol. 3:26–29.

Costanza, R. (1992). Toward an operational definition of ecosystem health. *Ecosystem health: New goals for environmental management*. 239–256.

den Hartigh, E., Tol, M., and Visscher, W. (2006). The health measurement of a business ecosystem. *Proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting*. 1–39.

Iansiti, M. and Levien, R. (2004a). The keystone advantage: What the new dynamics of business ecosystems mean for strategy, innovation, and sustainability: Harvard business school press. *Boston, MA*.

Iansiti, M. and Levien, R. (2004b). Strategy as ecology. *Harvard business review*, vol. 82(3):68–81.

Lóscio, B. F., Burle, C., and Calegari, N. (2017). Data on the Web Best Practices. `https://www.w3.org/TR/dwbp/`. Acesso em: 01/05/2018.

Manikas, K. and Hansen, K. M. (2013). Reviewing the health of software ecosystems–a conceptual framework proposal. *Proceedings of the 5th International Workshop on Software Ecosystems (IWSECO)*. 33–44.

Oliveira, M. I. S., Lima, G. d. F. A. B., and Lóscio, B. F. (2017). Investigations about data ecosystems: A systematic mapping study. *Submetido para Knowledge and Information Systems*.

Oliveira, M. I. S. and Lóscio, B. F. (2018). What is a data ecosystem? *Aceito em Digital Government Research*.

Schaeffer, D. J., Herricks, E. E., and Kerster, H. W. (1988). Ecosystem health: I. measuring ecosystem health. *Environmental Management*, vol. 12(4):445–455.

Von Alan, R. H., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS quarterly*, vol. 28(1):75–105.

Wahyudin, D., Mustofa, K., Schatten, A., Biffl, S., and Min Tjoa, A. (2007). Monitoring the "health" status of open source web-engineering projects. *International Journal of Web Information Systems*, vol. 3(1/2):116–139.

A Process for Reverse Engineering of Aggregate-Oriented NoSQL Databases with Emphasis on Geographic Data

Angelo Augusto Frozza[1,2], **Ronaldo dos Santos Mello**[1] **(advisor)**

[1]PPGCC-INE - Federal University of Santa Catarina (UFSC)
Florianópolis – SC – Brazil

[2]Federal Institute Catarinense (IFC)
Camboriú – SC – Brasil

angelo.frozza@ifc.edu.br, r.mello@ufsc.br

Level : Doctorate
Admission : March 2016
Qualification Exam : June 2018
Defense Forecast : February 2020

Completed Steps : Bibliographic review, Progress Seminar, Schemas extraction for document-oriented NoSQL DBs, Qualification exam.

Future Steps : Schema extraction for key-value and columnar NoSQL DBs, mapping JSON Schemas to RDF and validation.

Resumo. *Bancos de dados (BD) NoSQL não têm esquemas ou permitem esquemas flexíveis. Esta característica é adequada para vários domínios de aplicações, tais como, redes sociais, aplicações Web e Internet das Coisas. Porém, é crescente o interesse em manipular esquemas de BD NoSQL, p.ex. para tarefas como integração de BD. Este trabalho propõe um processo para extração de esquemas de BD NoSQL orientados a agregados (com possíveis dados geográficos) e a produção de visões RDF a partir dos esquemas criados. Pelo nosso conhecimento, não há trabalhos com uma abordagem tão abrangente ou que tratem dados NoSQL geográficos, demonstrando que ainda há muito a ser feito nessa área. A partir da análise dos trabalhos relacionados, apresentamos a nossa proposta e alguns resultados preliminares.*

Abstract. *One main advantage of NoSQL DBs is having no schemas, or allowing flexible ones. Such feature is suitable for various application domains, including social networks, Web applications, and Internet of Things. However, there is a growing interest in the management of NoSQL DB schemas, e.g., for tasks such as DB integration. This work proposes a process for the schema extracting from aggregate-oriented NoSQL DB (with possible geographic data), and the production of RDF views from the created JSON Schemas. To our knowledge, there are no related works that propose such a comprehensive approach and that deal with geographic NoSQL data. We analyze ten related works published in the last years, which present different proposals for NoSQL schema extraction, but few of them focus on the schema integration, demonstrating that there is still much to be done in this area. From this analysis, we present our proposal and some preliminary results.*

Keywords*: NoSQL. Data Reverse Engineering. Schema Extraction. Geographic data.*

1. Introduction

The *Big Data* market explosion has led large companies to demand databases (DB) that can handle large data volumes effectively. In this context, traditional Relational Databases (RDB) present several limitations, e.g., they prioritize strong consistency. The NoSQL DBs have emerged to deal with these RDB limitations [Cattell 2011]. They, generally, provide eventual data consistency, a robust availability and elasticity capabilities. A common feature of NoSQL DBs is that they are *schemaless*, i.e., they allow the storage of data without prior knowledge of their structure [Sadalage and Fowler 2013]. However, the lack of information regarding the schema makes difficult to perform several data processing tasks, such as data integration, data retrieval, validation, and analysis [Kapsammer et al. 2012]. Moreover, have information about the data scheme is very useful for application development. For example, several applications, such as *Foursquare*, retrieve data in JSON format but do not define a schema, being difficult for users to query such data because they are unaware of the documents structures. In short, a schema is useful because allows us to understand the data structure of a dataset.

Based on this motivation, the objective of this work is propose a process for the Reverse Engineering of NoSQL DBs that supports the aggregate-oriented data model, i.e., NoSQL DBs based on key-value, document-oriented and columnar data models. Our process presents several highlights, among them, the handling of complex data types (typical of NoSQL DBs) and of spatial data. The expected contributions are: (*i*) a process for generating aggregate-oriented NoSQL DB schemas; (*ii*) a canonical model based on JSON Schema recommendation and a process of mapping aggregate-oriented NoSQL schemas to the recommendation, considering an extension of the JSON Schema to represent geographic properties; (*iii*) a method for merge distinct schemas (represented in JSON Schema) from NoSQL DBs with heterogeneous aggregate-oriented data models and with possible geographical properties (*e.g.* point, line, polygon, multipoint, multiline, multipolygon, and geometry); (*iv*) generation and persistence of semantic visions in the RDF (Resource Definition Language)[1] format from the mapping of unified JSON Schemas, aiming to collaborate with applications that wish to make queries to multiple NoSQL DBs with possible geographic data.

The rest of this paper is organized as follows: Session 2 presents the problem statement and relevance; Section 3 presents the process proposed in this Thesis; Section 4 presents the preliminary results obtained with the extraction of NoSQL DB schemas, and Section 5 describes the future activities to be carried out to complete this Thesis.

2. Problem statement and relevance

When the database paradigm changes, new processes and tools need to be developed. So it was in the transition to RDB in the 1980s and 1990s; then, in the 2000s, with the growth in the use of XML data; and now with the NoSQL DBs.

In the NoSQL DBs context, the lack of schemas or the use of flexible schemas to define the data structure offers greater ease in dealing with scalability and increasing availability. However, this feature becomes a challenge for the data processing tasks because of the lack of homogeneity that it presents, which does not occur with RDBs. Another feature of schemaless DBs is to facilitate both the registration of complex and non-uniform

[1] https://www.w3.org/RDF/

data as the evolution of the data [Ruiz et al. 2015]. In general, common NoSQL data can be represented in JavaScript Object Notation (JSON)[2] format, while spatial data can be represented in GeoJSON[3] format. Both formats are recent industry standards. Although GeoJSON is a suitable standard for storing geographic data, in NoSQL this data type can be stored in several other formats, such as text, GML, KML, WKT, among others. Identifying geographic data stored in other data formats, different of GeoJSON, represents another challenge to be addressed in this research.

Despite these facilities, its increasingly perceived that a growing interest in the management of explicit NoSQL schemas is emerging [Klettke et al. 2015, Ruiz et al. 2015, Karpov 2017, Ruckstieß 2017]. However, there is not a standard for representing schemas for JSON data. The JSON Schema[4] is a vocabulary that is under discussion and is heading to become the default schema definition for JSON documents.

Most of the works analyzed (Table 1) aim to create tools for manipulate and manage schemas [Izquierdo and Cabot 2013, Kapsammer et al. 2012, Klettke et al. 2015, Mesiti and Valtolina 2014, Ruiz et al. 2015, Wang et al. 2015, Baazizi et al. 2017]. Two papers emphasize the use of JSON data in RDBs [Discala and Abadi 2016, Liu et al. 2016]. Only four papers presented a complete Reverse Engineering process. They extract schemas from several NoSQL datasets and produce an integrated conceptual schema [Kapsammer et al. 2012, Izquierdo and Cabot 2013, Kiran and Vijayakumar 2014, Mesiti and Valtolina 2014].

The works of [Kapsammer et al. 2012] and [Izquierdo and Cabot 2013] use a similar approach to identify schemas implicit in JSON documents coming from social networking APIs and Web applications. The schemas obtained are integrated to generate a domain schema for the application. The two works emphasize the extraction of schemas from JSON documents obtained through calls to application APIs and products a schema represented in the ECORE model[5]. In this Thesis, we intend to extract the schemas of NoSQL DBs with heterogeneous data models, unifying them in an RDF schema.

BigLoader [Mesiti and Valtolina 2014] is a theoretical design that aims to support the user in data loading activities in NoSQL DBs. The work proposed by [Kiran and Vijayakumar 2014] extracts schemas from tables stored in HBase that are used to create local ontologies in OWL (Web Ontology Language)[6]. These are used to form a global ontology, which can be used to perform searches from a SPARQL endpoint. We consider a limitation of this work that the schema extraction process only is applied to HBase. Our approach is more suitable, running the Reverse Engineering of any NoSQL DB that supports an aggregate data model.

Some works use an intermediate data model different from the model used to represent the final schema. This intermediate model is generally used in the initial stages of the process to represent the structure of a single NoSQL data instance. A hierarchical structure, used to represent the structure of nested JSON document objects, is proposed by [Klettke et al. 2015, Wang et al. 2015, Discala and Abadi 2016]. Works

[2]http://json.org/
[3]http://geojson.org/
[4]http://json-schema.org/
[5]https://www.eclipse.org/modeling/emf/
[6]https://www.w3.org/OWL/

Table 1. Comparing the works of Reverse Engineering from NoSQL DBs

Paper	Data Entry (Source)	Extraction or Integration Strategy	Intermediate Model	Data Types	Versions	Integration	Standards	Output
Kapsammer et al. 2012	**JSON** (social networks APIs)	MDE	**JSON Schema**	**Yes**	**Yes**	**Yes ***	**Yes**	ECORE *Schema*
Izquierdo et al. 2013	**JSON** (web services APIs)	MDE	JSON metamodel	**Yes**	**Yes**	**Yes ***	**Yes**	ECORE domain schema
Kiran et al. 2014	*HBase*	Hierarchical Summarization	*HBase*	No	**Yes**	**Yes**	**Yes**	**OWL Ontology**
Mesiti et al. 2014	**JSON** and others	Hierarchical Summarization	–x–	N/I	No	**Yes**	N/I	Key-value Schema
Klettke et al. 2015	**JSON** (MongoDB dataset)	Hierarchical Summarization	Structure Identification Graph (SG)	**Yes**	**Yes**	No	**Yes**	JSON Schema
Ruiz et al. 2015	**JSON** (MongoDB, CouchDB e HBase)	MDE	JSON metamodel	**Yes**	**Yes ****	No	**Yes**	NoSQL schema metamodel
Wang et al. 2015	**JSON** (Datasets)	Hierarchical Summarization	–x–	No	**Yes**	No	No	*eSiBu-Tree*
Discala et al. 2016	**JSON** (Dataset JSON or CSV)	Machine Learning	Directed graph	**Yes**	No	No	No	Relational Schema
Liu et al. 2016	**JSON** (Oracle JSON column)	Hierarchical Summarization	–x–	**Yes**	No	No	No	JSON DataGuide
Baazizi et al. 2017	**JSON** (Dataset)	Hierarchical Summarization	–x–	**Yes**	No	No	No	Proprietary Key-value Schema
This Thesis	**JSON (aggregate-oriented NoSQL)**	**Hierarchical Summarization and MDE**	**JSON Schema**	**Yes**	**Yes**	**Yes**	**Yes**	**RDF**

* Integration of schemas obtained from Web service APIs.
* Only work that produces versioned schemas.

based on MDE [Izquierdo and Cabot 2013, Ruiz et al. 2015] use ECORE metamodels. Works that cite the use of JSON schemas generally adopt *key-data_type* constructs. Only [Kapsammer et al. 2012] use the JSON Schema recommendation as an intermediate model. We propose use JSON Schema to store the schemas found in a NoSQL dataset.

Notice that some works are limited to analyze the structure of JSON documents, not doing any special treatment about the data types of each atomic element in the JSON document. We can also notice that just few approaches uses standards recommended by organizations such as W3C[7] to represent the final schema produced in the Reverse Engineering process. In this Thesis, we consider that standards such as JSON, GeoJSON, JSON Schema and RDF can contribute to solve the Reverse Engineering problem and also simplify the work by not requiring the developer to define new technologies or tools.

Regarding the final representation of the schema, there is also no consensus. The works that use the MDE approach present an application domain schema as an ECORE metamodel [Kapsammer et al. 2012, Izquierdo and Cabot 2013, Ruiz et al. 2015]. Other works like *eSiBu-Tree* [Wang et al. 2015] use their own structure. While [Baazizi et al. 2017] proposes the own schema language, based on JSON Schema, [Kiran and Vijayakumar 2014] adopts OWL as the final model of schema representation. Only [Klettke et al. 2015] use the JSON Schema recommendation. We propose to use RDF to represent an data schema and a catalog containing the mappings between the local JSON Schema and the schema view in RDF.

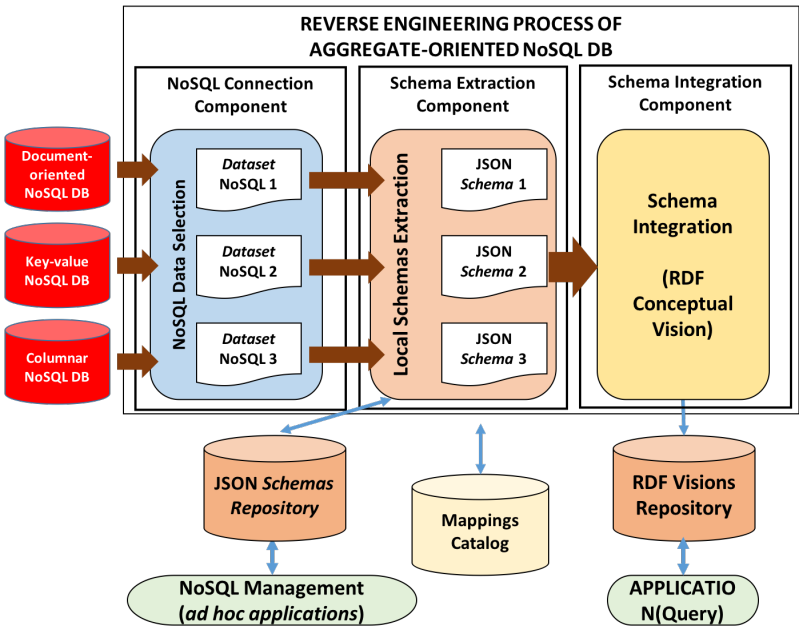
[7]https://www.w3.org/

Finally, we do not found any work emphasizing the spatial data Reverse Engineering in NoSQL DBs, this being another point of originality of this Thesis.

3. Proposed NoSQL Schema Extraction Methodology

Figure 1 presents an overview of the methodology for Reverse Engineering of NoSQL DBs proposed in this Thesis. It can serve as a basis for the construction of data integration frameworks running between NoSQL DBs and the applications.

Figure 1. Process for Reverse Engineering of aggregate-oriented NoSQL DBs



The Reverse Engineering process starts with the connection to the NoSQL DBs through the *NoSQL Connection Component*. This component can select the entire DB or a sample (filtered) data. NoSQL data is loaded considering the data model of each NoSQL DB. If geographic data are present, they can be loaded in GeoJSON or converted to this format if they are in another similar spatial data format.

The connection component delivers pre-selected NoSQL dataset to the *Schema Extraction Component*. This component parses the input NoSQL data to produce local schemas in the JSON Schema format. For each dataset, a single local schema must be generated, which is later stored in a *JSON Schemas Repository*. The extraction process analyzes the structure of the NoSQL data, as well as its attributes, to infer information such as JSON data types and mandatory/optional elements. This information extracted from the data structure is stored in a *Mapping Catalog* and can be used to enrich the conceptual view to be produced in the next step. Since the JSON Schema recommendation does not support the specification of spatial data schemas, an extension of this recommendation should be proposed to allow this specification. Once the local schemas are extracted in the JSON Schema format, possible with geographic properties, the *Schema Integration Component* takes effect. It aims to integrate the local schemas obtained previously and generate an unified semantic view represented in RDF. This integration process is based on matching operations of schemas appropriate to JSON Schema. The Schema

Integration Component will be developed in future works. In this Thesis, it is only intended to map the JSON Schemas, produced in the Schema Extraction Component, to a RDF schema representation.

The RDF Schema produced must be stored in an *RDF View Repository* so applications developers with interest in the integrated data can access it. Also, the Mapping Catalog should be populated with information about the matches between the local JSON Schemas and the RDF representation.

4. Experimental Evaluation

We conducted initial experiments to verify the correctness and completeness of the JSON $\rightarrow$ JSON Schema mappings. To do so, five JSON documents with several data types and heterogeneous structures were created. An accuracy of 100% was obtained since all expected data types in the input documents were identified. These data types include mandatory attributes, Extended JSON types, number of items in an array, and union. Experimental results showed that the processing time spent on reading the documents and generating the raw schemas reached around 99% of the total processing time, since all the documents in a collection must be read. It shows that the bottleneck of our process is the document reading and not the processing steps themselves.

We also evaluate our approach considering the same JSON dataset used by the works of [Wang et al. 2015] and [Baazizi et al. 2017]. These datasets have many different raw schemas, the number of raw schemas extracted is very close to the total of documents in the dataset. According to [Wang et al. 2015], this is due to the nature of the datasets, such as *DBPedia*, whose documents usually have some heterogeneities. The results points out that the accuracy of our approach is promising, being equivalent or superior than related work. This was because we consider the name and type of each document attribute to define the raw schema, and not only the attribute names. In fact, the data type is also important to allow more accurate queries with filters on certain attributes.

5. Methodology and plan

This paper presented a methodology for Reverse Engineering of aggregate-oriented NoSQL Databases, with emphasis on geographic data. Our methodology considers the element data types (JSON and Extended JSON data type), including geographic data, for creating schemas in the JSON Schema format. We also consider creating conceptual views of schemas in RDF format. Preliminary experiments have demonstrated that our methodology is equivalent to or superior to state-of-the-art approaches.

This Thesis is in the middle of its development, and the next steps are: a) update the works related to the state-of-the-art; b) development of local schema extraction prototypes for key-value and columnar NoSQL DBs; c) propose roles to mapping JSON Schemas to RDF format and to develop complementary experiments to validate the whole process; d) writing articles; e) defense of the Doctoral Thesis, scheduled for Feb. 2020.

Two papers with partial results were presented and awarded at the Regional School of Database (ERBD 2017 and 2018). An article was published at IRI 2018 (Qualis B1), addressing schema extraction in document-oriented NoSQL DB . Other publications planned are: *i*) a survey on NoSQL DB schema extraction to be submitted to the

ACM Computing Surveys (Qualis A1); *ii*) a paper describing the integration component to be submitted to the IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER) (Qualis A2); *iii*) a paper describing the Reverse Engineering process of Geographic Data and the use of GeoJSON and JSON Schema to be submitted to ACM Transactions in GIS (Qualis B1).

References

Baazizi, M.-A., Lahmar, H. B., Colazzo, D., Ghelli, G., and Sartiani, C. (2017). Schema Inference for Massive JSON Datasets. In *Proc. 20th EDBT*.

Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, 39(4):12.

Discala, M. and Abadi, D. J. (2016). Automatic Generation of Normalized Relational Schemas from Nested Key-Value Data. In *Proceedings SIGMOD '16*, pages 295–310, New York, NY, USA. ACM.

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. Springer-Verlag.

Kapsammer, E., Kusel, A., Mitsch, S., Proll, B., Retschitzegger, W., Schwinger, W., Schonbock, J., Wimmer, M., Wischenbart, M., and Lechner, S. (2012). User profile integration made easy - Model-driven extraction and transformation of social network schemas. In *Proc. 21st WWW*, pages 939–948.

Karpov, V. (2017). Mongoose NPM package.

Kiran, V. K. and Vijayakumar, R. (2014). Ontology based data integration of NoSQL datastores. In *9th ICIIS*, pages 1–6.

Klettke, M., Storl, U., and Scherzinger, S. (2015). Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores. In *BTW*, volume 241 of *LNI*, pages 425–444. GI.

Liu, Z. H., Hammerschmidt, B., Mcmahon, D., Lu, Y., and Chang, H. J. (2016). Closing the Functional and Performance Gap Between SQL and NoSQL. In *Proceedings of the SIGMOD'16*, pages 227–238, New York, NY, USA. ACM.

Mesiti, M. and Valtolina, S. (2014). Towards a user-friendly loading system for the analysis of big data in the internet of things. In *Proceedings IEEE 38th COMPSACW*, pages 312–317.

Ruckstieß, T. (2017). Mongodb-schema NPM package.

Ruiz, D. S., Morales, S. F., and Molina, J. G. (2015). Inferring versioned schemas from NoSQL databases and its applications. *Lecture Notes in Computer Science*, 9381:467–480.

Sadalage, P. J. and Fowler, M. (2013). *NoSQL distilled : a brief guide to the emerging world of polyglot persistence*. Addison-Wesley.

Wang, L., Hassanzadeh, O., Zhang, S., Shi, J., Jiao, L., Zou, J., and Wang, C. (2015). Schema Management for Document Stores. *VLDB Endowment*, 8(9):922–933.

Partitioning Very Large de Bruijn Graphs for Genome Assembly

Author: Julio O. Prieto Entenza[1]
Advisor: Sérgio Lifschitz[1]

[1]Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

{jentenza, sergio}@inf.puc-rio.br

Level: Doctoral Degree
Admission: March 2015
Expected Conclusion: February 2019

Abstract. *The genome assembly is a fundamental problem in bioinformatics. For de novo assembly procedures, without a reference genome, the de Bruijn graph is used to perform a computational evaluation. The datasets produced by current instruments can reach billions of reads and a few terabytes of data. As a consequence, the graph processing involves large data management techniques, such as distribution and parallelism, to allow feasible and efficient solutions. However, the level of parallelism in the construction of the contigs is limited because the graph traversal methods need to visit many partitions to create them. Thus, the consequence is a rise in the volume of communication between the processing nodes. This doctoral research proposes a new approach to decrease the communication during graph traversals in large volumes of data, through a partitioning according to the de Bruijn graph properties and their relationships to contigs.*

Resumo. *A montagem de fragmentos de genoma é um problema fundamental na bioinformática. Na montagem de novo, onde não existe uma cadeia de referência, é usada a estrutura de dados do grafo de Bruijn para realizar o processamento computacional. Os dados produzidos pelos instrumentos podem atingir bilhões de reads e alguns terabytes de dados. Consequentemente, o processamento do grafo envolve técnicas de gestão de grandes volumes de dados, como distribuição e paralelismo, permitindo soluções viáveis e eficientes. No entanto, o nível do paralelismo na construção dos contigs, que fazem parte do resultado da montagem, é limitado pois os algoritmos para percorrer o grafo podem visitar vários nós. Muitas soluções computacionais foram desenvolvidas para diminuir a comunicação entre os nós, fundamentalmente mecanismos de bloqueio e particionamento baseado em funções de hashing. Entretanto, é observado um aumento do volume de comunicação entre os nós de processamento. Este trabalho de pesquisa de doutorado propõe uma nova abordagem para diminuir a comunicação durante os percursos de grafos envolvendo grandes volumes de dados, em um particionamento de acordo com as propriedades do grafo de Bruijn e seu relacionamentos com os contigs.*

1. Introduction

Genome sequencing is the process of determining the order of nucleotides within a DNA molecule. Nowadays, it is a fundamental pillar for biological research, medical diagnosis, and biotechnology. However, current technologies for DNA sequencing cannot read whole genomes in a single run. Thus, it breaks a genome into a set of many short sequences (*reads*), string over the alphabet $\Sigma = \{A, C, G, T\}$, whose whole sequence is then reconstructed. The genome assembly problem consists in to combine these reads to reconstruct the original DNA sequences without a reference genome[Nagarajan and Pop 2009].

In practice, assemblers output several *contigs* which are an accurate reconstruction of a region from a DNA sequence. Therefore, the problem to solve is also named *contig assembly problem* [Simpson and Pop 2015]. For the resolution of the *contig* assembly problem, the main approaches are based on the *de Bruijn graph* (dBG). In this type of assembly, we break each read into a sequence of overlapping *k*-length substrings (*k-mers*). Later, we add the distinct *k-mers* as an edge links graph's vertices and those *k-mers* whose positions are adjacent in a read. We may formulate the assembly problem as finding a graph traversal that visits each edge in the graph once (Eulerian tour) or at least one (Chinese Postman tour) [Medvedev et al. 2007].

There is a growing gap between the output of the new generation of massively parallel sequencing machines and the ability to analyze the resulting data. The datasets produced by current instruments can reach billions of reads and represent hundreds of gigabytes or even terabytes of data [Schmidt and Hildebrandt 2017]. Dealing with this tremendous amount of information requires either to use substantial computational resources or to conceive specific algorithms and data structures designed for resource efficiency. As sequencing cost continues to decrease, sequencing very large genomes become affordable, but an assembly of such genome is barely possible. Consequently, to face the challenge to produce correct assemblies, the future assemblers will have to handle larger and larger datasets, to deal with large genomes or meta-genomes while providing a high throughput to follow the sequencing rate.

In the *de Bruijn* case, there are two main computational challenges, the size of the graph and the massive parallelization to process it. The size of the graph is a bottleneck because the memory cost is proportional to the genome size and complexity. If bacteria genomes take only a few gigabytes of RAM large genomes, such as mammalian and plants, requires over tens to hundreds of gigabytes. In the case of the human case with approximately 3 Gbps of genome size, we could have 4.8 billion nodes and 384 billion arcs. As a comparison, the Facebook's social graph had roughly 1.39 billion nodes with over 1 trillion edges in 2015 [Ching et al. 2015]. It should be noted that there are organisms or collection of them, (such as plants and metagenomes) with a genome much larger than the human genome. For instance, a 1,000 Genomes dataset with 200 terabytes of data can generate about 2^{47} *k-mers* (or nodes), 64-128 times larger than the problem size of the top result in the Graph 500 list [Meng et al. 2016].

On the other hand, several contributions were proposed to offer fast genome assembly through the use of massive multi-core servers. Since most assembly algorithms consist mostly of graph traversal operations, it is difficult to increase the throughput by optimizing these operations since they mainly rely on memory accesses. One of the main issues is that de Bruijn graph is sparse (e.g., for humans the dBG would be a 3×10^{-9} x

3×10^{-9} adjacency matrix with 2-8 non-zeros per row) and an extremely high diameter graph. As a result, the genome assembly computation is dominated by irregular memory access patterns and fine-grained synchronization. This situation leads to very high memory usage and more generally very high resource consumption systems.

Different solutions have been proposed to address the genome assembly problem. They vary from specific applications to distributed systems, NoSQL databases, and cloud computing [Sohn and Nam 2016]. Although their apparent differences, all of them have in common the use of graph systems to represent and process a dBG. Because of in the next future, the size of datasets will increase dramatically; this situation will stress the graph systems. Consequently, there is a need for new approaches to process all of this massive amount of information in a scalable way.

This work is structured as follows: we formalize next the technical challenges and the scientific problem (Section 2). Section 3 covers a description of some related works. We describe our proposed approach in Section 4. Later, a summarized methodology and the current state of the research are exposed in Section 5. Finally, Section 6 lists the expected contributions of our research work.

2. Problem statement

The challenge to enable massive parallelization is to limit the processes communications. The question of memory access is also critical when the massive parallelization requires the use of multiple machines communicating via very slow network accesses. If there is a partitioning function that could accurately predict how *k-mers* are placed into *contigs*, we could create a partitioning scheme in such way that we would map the *k-mers* belonging to the same *contig* to the same node. Thus, during the traversal, a processor would not incur in communication costs since none of the *k-mers* (e.g., lookups in a distributed hash table) that build up a particular *contig* are local to a single processor (local buckets in the distributed hash table). This idea is similar to general graph partitioning, which aims to minimize the number of edges between separated components. However, the main difficulty is that we do not know which are the *k-mers* that belong to a *contig*.

Therefore, we may define our **scientific problem** as *Is it possible to decrease the communication cost in a very large de Bruijn graph during the genome assembly process?* If yes, we could also reduce the overall cost of the *contig* generation process.

3. Related works

Despite the several works about dBG for genome assembly, the partitioning of the graph to decrease the communication cost has received inadequate attention. The standard approach is to distribute the *k-mers* evenly among the processors through a hash function [Jackman et al. 2017][Meng et al. 2016] to keep a well-balanced partitioning. Although this method is efficient, scalable and creates balanced partitions, it presents poor locality concerning *contigs*. The main problem with the hash approach is that it could map adjacent *k-mers* to different partitions. A critical property of adjacent *k-mers*, ignored by the hash functions, is that they share a common substring called *minimizer*. Thus, when parallel processes need to iterate over the *k-mers* to create *contigs*, they incur in higher communication costs, memory usage, and synchronization costs because they need to communicate to other partition where an adjacent *k-mer* exist.

Other approaches based on a minimizer have been proposed to bypass the hash problem by exploiting the ability of the minimizer to use sequence contiguity while binning the *k-mers*. The Minimum Substring Partition [Li et al. 2013] breaks the short reads into multiple small partitions based on a minimum p-substring of the *k-mers*. This partitioning allows consecutive *k-mers* to be distributed in the same partition and decreasing the number of I/O operations. Recently, BCALM2 [Chikhi et al. 2016] distribute the *k-mers* over disk partitions but with the direct construction of a compact dBG by the compaction of consecutive *k-mers* that constitute a simple path in the graph.

Although these approaches increase the locality through the minimizers for processing the dBG, their main drawback is that the same *k-mer* could be distributed (copied) on different partitions. The minimizer of a *k-mer* depends on its neighborhood which, in turn, depends on the read where the *k-mer* exists. Thus, when a processor selects a *k-mer*, it needs to communicate with different partitions to traverse the dBG. If we take into account that the number of unique *k-mers* can be in the orders of billions, then the number of cross-partition communication could be high.

4. Proposed approach

Before presenting our solution, we need to present some basic definitions and terminology involving graphs, particulary the *de Bruijn graph*.

A short read is a string over alphabet $\Sigma = \{A, C, G, T\}$. A *k-mer* is a string whose length is k. Let $G = (V, E)$ be a de Bruijn graph (dBG) with a set of vertices V and a set of edges E. Each vertex $u \in V$ represents exactly a *k-mer* and each edge $(u, v) \in E$ is a binary relation between two vertices such as $u \rightarrow v$ exists iff $suf_k(u) = pref_k(v)$. Let a *unitig* be a path $x_1, \ldots, x_m$ in the graph where all out- and in-degree of the nodes x_i for all $1 < i < m$ are equal to 1, and the in-degree of x_m and the out-degree of x_1 are 1. A *unitig* is said to be maximal if a vertex on either side can not extend it.

The l-minimizer of a string u is the lexicographical smallest l-mer that is substring of u [Li et al. 2013]. We define $lmin(u)$ (respectively $rmin(u)$) to be the l-minimizer of the $(k-1)$-prefix (respectively suffix) of u. If $u \rightarrow v$ then $rmin(u) = lmin(v)$ [Chikhi et al. 2016].

Two strings u and v are compactable in a set K if $u \rightarrow v$ and u is the only in-neighbor of v , and v is the only out-neighbor of u.

Methodology

The intuition behind our solution is based on a graph partitioning where each *k-mer* that belong to the same *contig* should be in the same partition. Despite the vast works done in graph partitioning [Buluç et al. 2016], in our case, the main issue is to find out how to determine if two k-mers belong to the same *contig*. However, we do not know the *contigs* a priori. Consequently, we cannot also know which set of *k-mers* belongs to a *contig*. We must remember that finding the set of *contigs* is the goal of an assembly algorithm.

To obtain a solution to our problem, let us try to draw on the intuition from an example. Considering Figure 1, we can see that any edge-covering graph traversal should visit C (or any other of the labeled traversals, for that matter). After all, once a traversal enters the starting point of C, it has no other choice but to continue all the way through

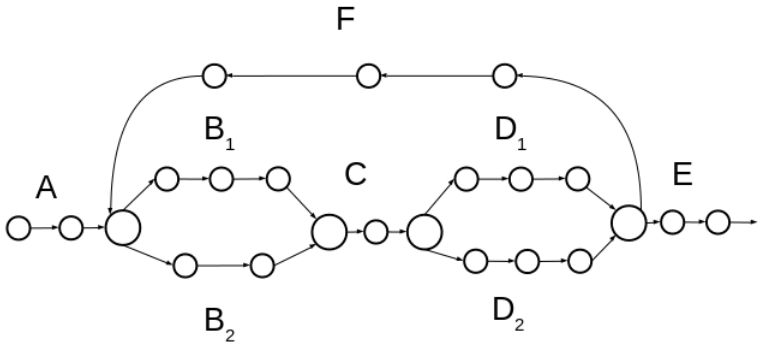


Figura 1. An example of a de Bruijn graph

the end of C. Remember that a *unitig* is a path such that all vertices except the first one have one incoming edge, and all vertices except the last one have one outgoing edge. As a maximal *unitig* is one that cannot be extended in either direction, the maximal *unitig* of a graph is a partition of the vertex set. Based on this intuition, we can define the algorithm, which outputs the partitions based on all the maximal *unitigs* in the graph.

A straightforward solution could use the BCALM2 method [Chikhi et al. 2016] to create the *unitigs* and, later, to make a partitioning over this set. However, if we apply this solution directly, we can lose the topological relationships among the maximal *unitigs* because these approaches ignore the nodes with more than two neighbors.

Consequently, our technique extends the BCALM2 approach and consists of three phases: 1) clustering according to the minimizers 2) contraction of the graph according to the *unitigs* and 3) cluster in the merge phase.

Algorithm 1 General algorithm

1: **for all** $x \in K$ **do in parallel**
2: Write x to $F(lmm(x))$
3: **if** $lmm(x) \neq rmm(x)$ **then**
4: Write x to $F(rmm(x))$
5: **end if**
6: **end for**
7: **for all** $i \in \{1, \ldots, 4^l\}$ **do in parallel**
8: Run $ContractBucket(i)$
9: **end for**
10: $\{\delta$ user defined threshold$\}$
11: $MergeComponents(\delta)$

Algorithm 2 $ContractBucket(i)$

1: Load $F(i)$ into memory.
2: $U \leftarrow i$-compaction of $F(i)$
3: $\forall u \in U$ initialize a forest $T = \{u\}$
4: **repeat**
5: Find t_i and t_j who have a common minimizer
6: $t_k \leftarrow t_i \cup t_j$
7: delete t_i and t_j
8: $T \leftarrow T \cup \{t_k\}$
9: **until** there is not elements to reduce

In the first stage, the algorithm distributes the k-mers (K) to files $F(1), \ldots, F(4^l)$. These are called bucket files. Each k-mer $x \in K$ goes into file $F(lmm(x))$, and if $lmm(x) \neq rmm(x)$, also in $F(rmm(x))$ to avoid false *unitigs* creation. The parameter l controls the minimizer size. This process is the same with BCALM2 [Chikhi et al. 2016].

In the second stage of the algorithm, we process each bucket file using the *ContractBucket* procedure. After the *k-mer* distribution of the first stage, the bucket file $F(i)$ contains all the k-mers whose left or right minimizer is i. Therefore, we can load $F(i)$

into memory and perform i-compaction and reduction on it. Then, we need to unite the different components within each partition because they represent partial *contigs*. We find all the branch nodes and add all the compacted edges to the forest. We repeat a similar process of finding the incident edge from each tree constructed so far to a different tree, and adding all of those edges to the forest. When it does, the set of edges forms a spanning forest. Since the size of the bucket is small, this compaction and reduction can be performed using in-memory algorithms. The resulting strings are then written to disk and will be processed during the third stage. At the end of the second stage, when all *ContractBucket* procedures are finished, we have performed all the necessary compressions and reductions on the data.

Algorithm 3 $MergeComponents(\delta)$

1: $\forall T$ initialize cluster $c = \{T\}$
2: **repeat**
3: Find c_i and c_j who share the same *k-mer*
4: $c_k \leftarrow c_i \cup c_j$
5: delete c_i and c_j
6: $C \leftarrow C \cup \{c_k\}$
7: **until** $|C| < \delta$

At this stage of the algorithm, the *k-mers* $x \in K$ with $lmm(x) \neq rmm(x)$ exist in two copies. These *k-mers* are always at the ends (prefix or suffix) of the compacted strings, never internal, and they can be recognized by the fact that the minimizer at that end does not correspond to the bucket where it resides [Chikhi et al. 2016]. As we have generated more components (forest) than the number of computers in the 2nd phase, we choose an adjacent cluster and pack it into the same container during the merge algorithm, We keep a small number of cross-cluster edges and double *k-mers*.

This method resembles a hierarchical agglomerative clustering [Clauset et al. 2004] because we build the hierarchy bottom up. Hence, we find two adjacent clusters and merge them if the number of double *k-mers* decrease and the number of *unitigs*, within the new cluster, increase. Then, we repeatedly merge two new pairs of clusters until there is no adjacent cluster, or we reach the number of partitions.

5. Methods

The thesis subject arose from a demand for assembling sugarcane sequences as a part of research cooperation between PUC-Rio's BioBD Laboratory and the Laboratory of Molecular Biology of Plants (IBqM), Institute of Medical Biochemistry at UFRJ. One goal is to study the sugarcane genome into Brazilian species, which is very complicated as there are high rates of heterozygosity and repetitions, demanding a high availability of main memory.

An initial bibliographical survey has been carried out looking for ad-hoc data structures, partitioning and clustering approaches that impact parallel dBG Traversal algorithms. Also, three assembling tools Abyss [Jackman et al. 2017], MSP [Li et al. 2013] and BCALM2 [Chikhi et al. 2016] were thoroughly studied, to understand how the k-mers can be allocated to preserve the *contig* relationship.

All of these activities allows one to have more in-depth knowledge and understanding of the problem, defining the variables involved besides some bounds and limits.

6. Contributions

This thesis proposal will bring a set of additional contributions as listed below:

- Identification and formalization of the main variables that impact the parallel graph traversal algorithms on dBG assemblers.
- A new approach of dBG partitioning based on minimizers.
- A survey of dBG assemblers, emphasizing the approaches on parallel and distributed traversal algorithms used.
- An actual implementation of the proposed approach.

Referências

Buluç, A., Meyerhenke, H., Safro, I., Sanders, P., and Schulz, C. (2016). Recent advances in graph partitioning. In *Algorithm Engineering*, pages 117–158. Springer.

Chikhi, R., Limasset, A., and Medvedev, P. (2016). Compacting de bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics*.

Ching, A., Edunov, S., Kabiljo, M., Logothetis, D., and Muthukrishnan, S. (2015). One trillion edges: Graph processing at facebook-scale. *Proceedings of the VLDB Endowment*, 8(12):1804–1815.

Clauset, A., Newman, M. E., and Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Jackman, S. D., Vandervalk, B. P., Mohamadi, H., Chu, J., Yeo, S., Hammond, S. A., Jahesh, G., Khan, H., Coombe, L., Warren, R. L., and Birol, I. (2017). ABySS 2.0: resource-efficient assembly of large genomes using a Bloom filter. *Genome Research*.

Li, Y., Kamousi, P., Han, F., Yang, S., Yan, X., and Suri, S. (2013). Memory efficient minimum substring partitioning. In *Proceedings of the VLDB Endowment*, volume 6, pages 169–180. VLDB Endowment.

Medvedev, P., Georgiou, K., Myers, G., and Brudno, M. (2007). Computability of models for sequence assembly. In *International Workshop on Algorithms in Bioinformatics*, pages 289–301. Springer.

Meng, J., Seo, S., Balaji, P., Wei, Y., Wang, B., and Feng, S. (2016). Swap-assembler 2: Optimization of de novo genome assembler at extreme scale. In *Parallel Processing (ICPP), 2016 45th International Conference on*, pages 195–204. IEEE.

Nagarajan, N. and Pop, M. (2009). Parametric complexity of sequence assembly: theory and applications to next generation sequencing. *J Comput Biol.*

Schmidt, B. and Hildebrandt, A. (2017). Next-generation sequencing: big data meets high performance computing. *Drug Discovery Today*, 22(4):712 – 717.

Simpson, J. T. and Pop, M. (2015). The theory and practice of genome sequence assembly. *Annual review of genomics and human genetics*, 16:153–172.

Sohn, J.-i. and Nam, J.-W. (2016). The present and future of de novo whole-genome assembly. *Briefings in bioinformatics*, 19(1):23–40.

Author Index

Almeida, Ana Carolina 29
Amorim, Fernanda Araujo Baião 53
Araújo B., F. Ronald 35
Arruda, Narciso 41
Becker, Karin 60, 81
Bedo, Marcos V.N. 5, 17
Brayner, Angelo 35, 41
Castilho, Francisco San Diego 41
Ciferri, Cristina Dutra de Aguiar 23
Duarte, Denio 95
Entenza, Julio O. Prieto 116
Fabro, Marcos Didonet Del 11
Ferranti, Nicolas 47
Frozza, Angelo Augusto 109
Galante, Renata 60
Grzeça, Marcos A. 60
Guedes, Thaylon 5, 17
Haeusler, Edward Hermann 29
Jasbick, Daniel L. 5
Kuszera, Evandro Miguel 11
Lifschitz, Sérgio 29, 116
Lima, Glória de Fátima Andrade Barros 102
Lóscio, Bernadette Farias 67, 74, 102
Machado, Nielsen Luiz Rechia 88
Madeiro, João Paulo 41
Mattoso, Marta 17
Mello, Ronaldo dos Santos 95, 109
Monteiro, José Maria 35, 41
Moura, Edleno 1
Muniz, Rayelle Ingrid Vera Cruz Silva 67
Oliveira, Daniel de 5, 17
Oliveira, Marcelo Iury S. 102
Oliveira, Pericles de 1
Oliveira, Rafael Pereira de 29
Oliveira, Rodolfo A. 5
Perciliano, Luciana de Sá Silva 53
Peres, Leticia M. 11
Rocha, Guilherme Muzzi da 23
Rodrigues, Daniel 41
Ruiz, Duncan Dubugras Alcoba 88
Santos, Gilberto 1
Santos, Lúcio F.D. 5
Santos, Wilker Cavalcante do Rego 74
Schreiner, Geomar André 95
Schwabe, Daniel 29
Silva, Altigran da 1
Silva, Vítor 17
Soares, Stênio Sã Rosário Furtado 47
Souza, Jairo F. De 47
Teixeira, Tibet 41
Torquato, Douglas 41
Vidal, Vânia 41
Vinuto, Tiago 41
Walter, Roberto 81