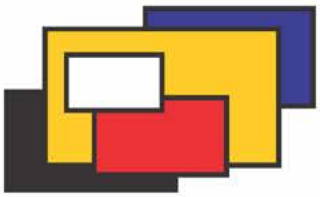


SBBD ONLINE 2020

35th Brazilian Symposium on Data Bases

De 28 de setembro a 2 de outubro de 2020
September 28th / October 2nd

Companion Proceedings

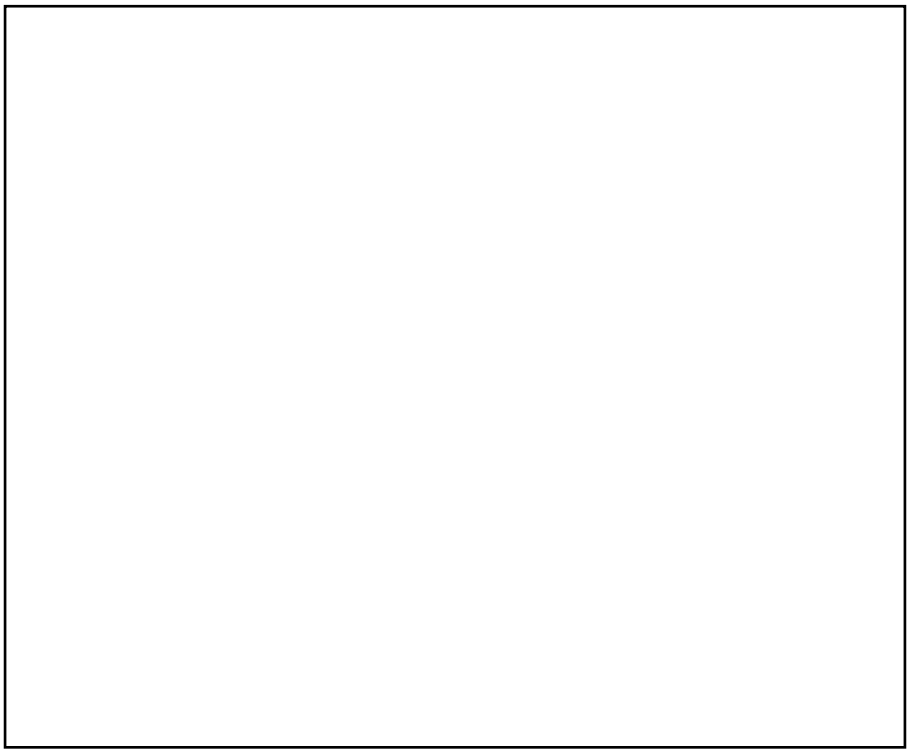
WTDBD - Workshop de Teses e Dissertações
WTDBD - Ongoing Graduate Research Workshop

Sessão de Ferramentas
Demonstrantions Sessions

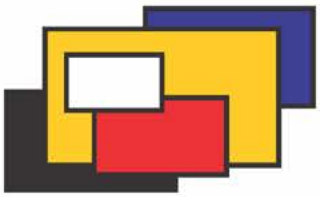
Tutoriais
Tutorials

DEPARTAMENTO DE INFORMÁTICA
PUC-RIO

CIP



ISSN:



SBBD ONLINE
2020

SBBD Organization

Program Chair
Fábio Porto (LNCC, Brazil)

Full Papers Co-chair
Daniel Oliveira (UFF, Brazil)
Short Vision Industrial Chair
Ricardo Torres (NTNU, Norway)
Steering Committee Chair
Carina Dorneles (UFSC, Brazil)
Demos and Applications Chair
Denio Duarte (UFFS, Brazil)
(WTDBD) Thesis and Dissertation Workshop Chair
Carlos Eduardo Santos Pires (UFCG, Brazil)
Short Courses Chair
José Maria Monteiro (UFC, Brazil)

ONLINE ORGANIZATION

SBBD General Chair
Sérgio Lifschitz (PUC-Rio)

Editorial

O Simpósio Brasileiro de Bancos de Dados (SBBD) é um dos eventos mais tradicionais da Sociedade Brasileira da Computação (SBC). Neste ano de 2020 comemoramos 35 anos de atividades e importantes contribuições científicas e acadêmicas. O SBBD envolve profissionais diversos, pesquisadores, professores, alunos de pós-graduação e graduação, uma comunidade interessada na área de Bancos de Dados, hoje ampliada com termos como Engenharia e Ciência de Dados. Por conta da pandemia do Covid-19, o evento foi organizado inteiramente online, entre os dias 28 de setembro e 01 de outubro. Se por um lado não tivemos as conversas e troca de experiências dos eventos presenciais, por outro lado foi possível uma maior participação, no Brasil e no exterior, ampliando o alcance do SBBD significativamente. Os Anais Estendidos do SBBD 2020 contêm resumos dos excelentes tutoriais convidados, apresentados por pesquisadores de renome internacional, como Altigran Silva, Amr el Abbadi e Patrick Valduriez. Apresentamos também os artigos selecionados e apresentados em eventos co-localizados e também já tradicionais, como é o caso da Sessão de Demonstração de Ferramentas (Demos) e o Workshop de Teses e Dissertações em Bancos de Dados (WTDBD). Estes eventos foram coordenados, respectivamente, pelos Professores Denio Duarte (UFFS) e Carlos Eduardo Santos Pires (UFCG). As apresentações estão todas gravadas e disponibilizadas no site do SBBD 2020 (http://sbbd.org.br/2020/) e acompanham estes anais como registro da qualidade dos eventos. Aproveitem!

Sérgio Lifschitz
Departamento de Informática, PUC-Rio
Coordenador Geral do SBBD2020 e Editor dos Anais Estendidos

The Brazilian Symposium on Databases (SBBD) is one of the most traditional events of the Brazilian Computer Society (SBC). In this year of 2020 we celebrate 35 years of activities, with strong scientific and academic contributions. The SBBD involves a community of professionals, researchers, professors, graduate and undergraduate students, all interested in the area of Databases, now expanded with terms such as Data Engineering and Data Science. Due to the Covid-19 pandemic, the event was organized entirely online, between September 28th and October 1st. If, on the one hand, we did not have the networking experiences of the face-to-face events, on the other hand, greater participation was possible, in Brazil and abroad, significantly expanding the SBBD's reach. The SBBD 2020 Companion Proceedings contain summaries of the excellent invited tutorials, presented by internationally renowned researchers such as Altigran Silva, Amr el Abbadi and Patrick Valduriez. We also present the selected articles presented in co-located and also traditional events, such as the Tools Demonstration Session (Demos) and the Database Thesis and Dissertation Workshop (WTDBD). These events were coordinated, respectively, by Professors Denio Duarte (UFFS) and Carlos Eduardo Santos Pires (UFCG). The presentations are all recorded and made available on the SBBD 2020 website (http://sbbd.org.br/2020/) and accompany these proceedings as a souvenir of the quality of the events. Enjoy!

Sérgio Lifschitz
Informatics Department, PUC-Rio
SBBD2020 General Coordinator and Companion Proceedings Editor

Sumário

WTDBD - Workshop de Teses e Dissertações

WTDBD - Ongoing Graduate Research Workshop

Editorial

The Workshop of Thesis and Master Dissertations in Databases (WTDBD) is a traditional event co-located with the Brazilian Symposium on Databases (SBBD). Due to COVID-19 and coronavirus pandemic, this year all activities of the event are online. The event gathers professor and graduate students from different Universities in Brazil to present and discuss their most recent database research results.

The WTDBD is an excellent opportunity to receive feedback upon on-going graduate work from experienced researchers. All submitted papers received four 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 seven accepted works (five masters and two doctorate works) from many different universities in Brazil.

The 2020 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 and the group of researchers that engaged with all their hearts in this endeavor. 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 2020 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.

Carlos Eduardo Santos Pires, UFCG
WTDBD 2020 - CP Chair

Detection of Depression Symptoms using Social Media Data

Silas P. Lima Filho [1,], **Jonice Oliveira**[1], **Monica Ferreira da Silva**[1]

[1]PPGI, Universidade Federal do Rio de Janeiro (UFRJ)
Av Athos da Silveira, 174 – 21941-916 – Rio de Janeiro – RJ – Brazil

silasfilho@ufrj.br, jonice@dcc.ufrj.br, monica.silva@ppgi.ufrj.br

Resumo. *Depressão é um das doenças que mais desabilita no mundo. Ela aflige pessoas de diferentes idades, gêneros e raças. A tarefa de identificar previamente os sintomas de depressão poderia ajudar tanto os profissionais como também pacientes em potencial a se aproximarem. Por conta do crescimento no uso das mídias sociais, mais pessoas estão procurando informações sobre saúde, doenças e tratamentos. Depressão não é uma exceção. Esse artigo apresenta uma proposta de método para detectar os sintomas de depressão no conteúdo de mídias sociais de um determinado usuário. Embora seja um trabalho em construção, nós apresentamos nossa proposta comparando com outras abordagens da literatura e também nos apoiamos na metodologia de Design Science Research e experimentos iniciais.*

Abstract. *Depression is one of the three leading disabling diseases. It affects people of different ages, gender and social classes. Identification of depression symptoms may help potential patients to get to the professionals in a shorter time, increasing recovery chances. Because of digital social media growth, more people are exchanging information about health, diseases, and treatment. Depression is not an exception. This article presents a method to detect depression symptoms by analyzing social media content generated by a user. Although it is under construction, we present our proposal in comparison with other literature approaches and show preliminary experiments. The research is structured based on DSR methodology.*

Admission on the program: 03/2017

Defense Date Expectation: 06/2021

Reference Dates: Qualification Exam (Late October - to be defined yet)

1. Introduction

Depression is one of the most reported mental diseases in the world. Some people call it the century illness due to its dangerousness[1]. The Global Burden of Disease indicates depressive disorders as the third lead cause of disability [James et al. 2018]. The World Health Organization (WHO) presents that around 300 million people from different ages suffer from some level of depression[2]. The Health Ministry in Brazil presents that 11.5 million people are affected by depression[3]. The task of identify people in the early phase of depression can face impediments like cost, social prejudice, and even a personal obstruction. [Lech et al. 2014] highlights the urgency in early identification and prediction of depression and its symptoms due to the difficulties to detect these symptoms in initial stages.

Infodemiology and *digital disease detection* are correlated terms to describe the use of digital platforms and tools to improve social health. They can be translated as efforts to tackle epidemics, identify individuals at risk, and communicate candidate urgent illness. The use of technology directly supports institutions, professionals, and even aids people to make themselves aware of some diseases [Horvitz and Mulligan 2015].

Social media has been used as online platforms to publish user's social interests and preferences. [Elkin 2008] presents that 34% of health search is made on social media, and 59% of adults look for health information on the internet. Therefore the content from these platforms can be seen as source of information that could help when dealing with disease detection or connection among a psychologist and one depressive patient. Due to the plenty of data offering, select what is the most effective, precise and representative data can be challenging. Therefore, choose a reliable technique and a consistent method analysis can require a great amount of research.

This paper aims to propose a new method to identify depression symptoms in social media platforms. Relied on the literature review, we believe this proposal highlights a research approach that was not investigated extensively. Although it is an in-progress research, with its stages and processes under validation, the following sections try to systematize the methodological steps to understand depression phenomena and how it affects people on social media.

On Section 2 we define the theoretical basis of the phenomena. Section 3 presents the basis for methodology and Section 4 depicts the proposed method. Finally, at Section 5 we conclude with expected contributions for this thesis research.

2. Depression on Social Media

For the literature selection, we have applied a systematic literature review **(SLR)** in order to have a deeper insight from the most recent research that tackles depression detection in social media. SLR allows to create protocols that can be reused by other researchers and therefore give to research transparency and reproducibility. We have based our effort on [Nakagawa et al. 2017] approach. This stage is under construction yet and it is intended to include two more bases. The SLR until this moment was done searching for articles in

[1] www.theguardian.com/news/2018/jun/04/what-is-depression-and-why-is-it-rising
[2] www.who.int/en/news-room/fact-sheets/detail/mental-disorders
[3] https://bit.ly/2YTsYHh

ACM and IEEE bases. It has been searched the string *("Social Media" OR "Social Network" OR "Complex Network") AND (Depression OR "Major Depressive Disorder").* Including only works from 2013 until 2018, from computing area which have used social media as a data source. The inclusion and exclusion criteria are listed below in Table 1. At the final stage, there was a total number of 47 selected papers. There were 22 papers from ACM Library and 25 papers from IEEE Explore. We list as follow contributions returned from SLR that summarizes part of the whole set of papers.

Inclusion	Exclusion
Directly tackles depression	Out of 2013-2018 scope
Have computational approach	Not written in english or portuguese
Attend both approaches	It is not a primary study
-	It does not have abstract
-	It does not have computing contribution
-	It has less than 4 pages

Table 1. SLR Criterias for inclusion and exclusion.

A good amount of articles relies on natural language processing (NLP) to make a systemic analysis over the text in social media publications. Not all the analyzed researches take into account the psychology point of view. The effect of taking into account existing approaches from psychology is that the analysis will be more robust and reliable since the psychology research area already addresses mental disease problems. It is a challenge align quantification made by metrics e.g. NLP, social network analysis and other techniques to the cognition of a psychologist on ordinary clinical treatment.

[De Choudhury et al. 2013] has developed many articles and researches about the measurement of depression in population using social media information. The authors in this work have been made use of psychometrics questionnaires. Psychometrics represents the theory and technique of measuring mental processes and it is applied in Psychology and Education. Similar to previous work, Tsugawa et al [Tsugawa et al. 2015] have applied the same analysis to replicate the results in a group of users from Japan.

[Park et al. 2015] present how activities on Facebook are associated with depressive states of users in order to raise awareness to depression at the University where the study was conducted, which had seen an increase in the suicide rate of its students. [Andalibi et al. 2017] explore self-disclosures posts in Instagram. In this article, the authors have used content from posts tagged with #depression to understand what rather sensitive disclosures do people make on Instagram. The work in [Li et al. 2016] is a qualitative study that tries to understand how is the behavior and comprehension of the Chinese population about depression. It is a qualitative study and differs from prior ones. [Vedula and Parthasarathy 2017] conduct an observational study to understand the interactions between clinically depressed users and their ego-network when contrasted with a group of users without depression. They identify relevant linguistic and emotional signals from social media exchanges to detect symptomatic cues of depression. [Chen et al. 2018] detected eight basic emotions and calculated the overall intensity (strength score) of the emotions extracted from all past tweets of each user. After that, they have generated a time series for each emotion of every user in order to generate a selection of descriptive statistics for this time series.

Papers cited above not always take into account how psychologists infer if someone is depressive or not. We also stress that many of the real contributions rely on textual information generated by one user. Since one of the depression symptoms in ICD 11[Association et al. 2013] is the inactivity, we could question if a depressive one would consistently generate online content. The context of psychology regularly deals with the subjectivity of information. Relied on that, we believe that relevant information can be extracted from other methods rather than text content. We believe that the classification of potential depressive users could be more reliable if combined with "subjective information".

3. Methodology

To achieve the objective of identifying depression symptoms, we rely on *Design Science Research* concept (DSR) due to its effort to systematize information technology research. The concept of DSR can have variations on its interpretation, although the authors in this area suggest the creation of artifacts as an approach to create solutions for certain types of research investigation. [Wieringa 2014] states DSR as *"...the design and investigation of artifacts in context"*.

Based on the above statement, it is suggested by [Hevner 2007] a three cycle model in order to create such solution. The model comprehends the *Relevance*, *Design* and *Rigor* cycles. The first one aims to find the needs of the observed context. The following cycle aims to create the artifacts that should mitigate the identified problem. The last cycle aims to support the artifact with identification of prior science knowledge. [Peffers et al. 2007] suggests a different workflow for the artifact creation. Even though the cited models seems different, they all suggest the construction of artifacts and its consequences in the studied environment. Therefore, based on [Peffers et al. 2007], we suggest the stages for this methodology proposal on Figure 1.

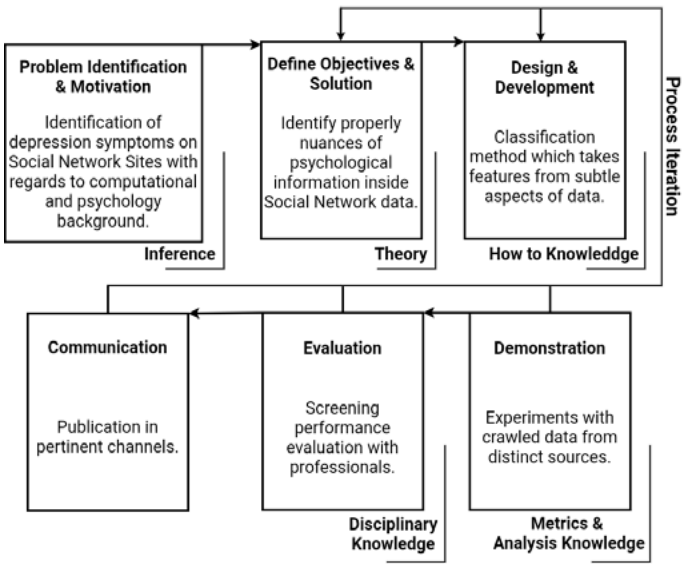


Figure 1. Methodology steps based on [Peffers et al. 2007] approach.

4. Proposal

The proposed solution, in order to create an artifact, faces different stages that can offer challenges to the research to be accomplished. Since the final product is the hability to

identify depression symptoms. We must pass by stages like obtain, process, model and persist the data from social network sites in a structure that can allow the further steps to retrieve this data for future analysis. Our method aims to identify a depressive user inserted in a social network in a reliable, consistent and unobtrusive manner. Figure 2 depicts how we intend to achieve it. Module 1 represents the phase where we construct the datasets to be explored in initial and later stages. This module is constructed mainly by crawlers and provides us depressive user's information concerning their vocabulary, content behavior and their preferences about relationships with other users. Afterwards datasets construction, through data modeling, Model 1 provides to Module 2 datasets from websites where depression is a key topic. The second module comprehends the data analysis and it splits this process into two substages. Both substages intend to abstract the implicit and explicit user's content. We represent Modules 1 and 2 as a cyclic process mainly due to the improvement of analysis, and evolution of datasets concerning length, complexity, and more abundat information.

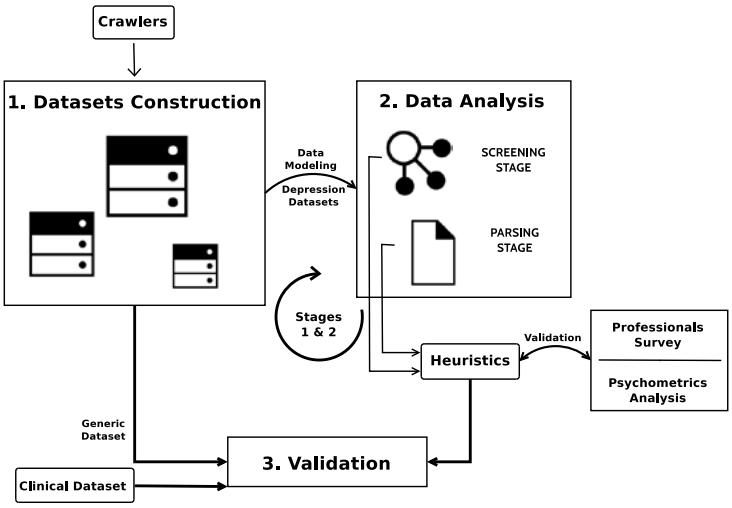


Figure 2. Conceptual schema of our proposal.

The screening substage comprehends a set of metrics that rely on topological analysis. Through social network analysis, we can understand how entities from a social network are connected. The topology study, through the use of social network analysis, can lead to the choice of specific users to have a better comprehension of them and their influence [Razis et al. 2020]. We can list as examples three different types of relevant analysis, *Social Influence Analysis*, *Node Classification* and *Community Detection* [Aggarwal 2011]. With these analysis, it would be possible to have a better understanding of the connections around a depressive person.

Parsing substage is divided into three sub-processes and it is represented in Figure 3. As most of the presented articles in Section 2, we also intend to apply text analysis over the datasets. Based on the work of [Sousa 2016], we aim to identify what are the main topics of each user in a social media platform. After the topic identification, for each user, we will find the correlation among topics and will map the terms into a polarity graph. The polarity graph can help to identify if the most used words and terms of a user tend to be positive or negative. Related work has shown that depressive people use to manipulate more negative words. This reflects the low self-vision of this group. With the polarity

graphs of each user, the third stage intends to analyze how these graphs have evolved. In this manner, it will be possible to check if someone's discourse has been turned into more positive or negative. The time series analysis could give clues about a tendency in user discourse content. One of the depression symptoms is the persistence of negative mood. Thus, one of possibilities is correlate the evolving of a polarity graph from topics to the tendency of negative humor of an user.

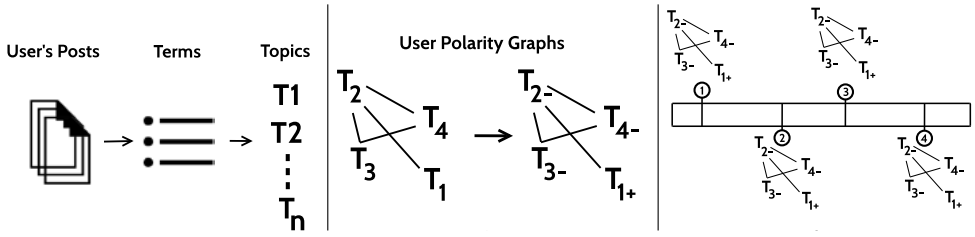


Figure 3. The Parsing substage's phases. Respectively, extraction of main topics from user content; Polarity of discovered topics; Evolution of polarity graphs over time.

As the iterations progress, Module 2 constructs a set of heuristics extracted from analysis. These heuristics are tested by professionals survey and compared to psychometrics. The heuristics set is also validated in Module 3. In this module, we intend to validate the heuristics not only by applying them in a more generic and wider dataset but also by testing in a psychology clinical scenario. In that way, we intend to corroborate the dynamics and behavior from the first iteration which was abstracted from particular datasets.

5. Expected Contributions and Future Steps

This work has presented an application scenario of informatics in mental health. Due to the wide implications of depression in people's health, and because of the great use of social media, we should use these data with better precision and reliability. We stress that Social Network Analysis (SNA) discerns the topological structure. Therefore, the main difference from prior research and our proposal, is to use psychology knowledge and ideas to improve classification performance with SNA and textual analysis. For the experimental phase, we already developed two collectors to obtain data from *HealingWell* and *Reddit* websites. Since this is an academic research, we do not have any intentions of publishing authors names and their personal information. We have at the moment a dataset with around 1.5 GB of collected information. The first dataset is related to the depression forum from HealingWell website[4]. We have collected 3075 posted topics and their respective posts summing a total of 18450 replying posts. With this experiment, we would understand the environment where the disease depression is the main topic of discussion, consequently, we would reproduce and validate the results and behavior of conversations about depression.

This work is still developing the methodology and the analysis that could be performed. The expected contributions are related to computer science and psychology. For the computer science, we expect that employing computational metrics like social network analysis and topic classification could support people to acquire a more accurate

[4]www.healingwell.com/community/default.aspx?f=19

understanding of how the phenomena of depression happen in social media and also how the social media reflects the real life. For the health research point of view (psychology, medicine), our approach could improve how the diagnosis of depression is performed. This could aid people to enjoy better health through the use of technology.

References

Aggarwal, C. C. (2011). *Social Network Data Analytics*. Springer Publishing Company, Incorporated, 1st edition.

Andalibi, N., Ozturk, P., and Forte, A. (2017). Sensitive Self-disclosures, Responses, and Social Support on Instagram: The Case of #Depression. In *Proc. of 2017 ACM Conf. on Comput. Supported Cooperative Work and Social Comput.*, pages 1485–1500, Portland, OR, USA. ACM.

Association, A. P. et al. (2013). *Diagnostic and statistical manual of mental disorders (DSM-5®)*. American Psychiatric Pub.

Chen, X., Sykora, M. D., Jackson, T. W., and Elayan, S. (2018). What about Mood Swings. In *Companion Proc. of the The Web Conf. 2018*, pages 1653–1660, New York, New York, USA. Int. World Wide Web Conf. Steering Committee.

De Choudhury, M., Counts, S., and Horvitz, E. (2013). Social Media As a Measurement Tool of Depression in Populations. In *Proc. of the 5th Annu. ACM Web Sci. Conf.*, WebSci '13, pages 47–56, New York, NY, USA. ACM.

Elkin, N. (2008). How america searches: Health and wellness. *Opinion Research Corporation: iCrossing*, pages 1–17.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4.

Horvitz, E. and Mulligan, D. (2015). Data, privacy, and the greater good. *Science*, 349(6245):253–255.

James, S. L., Abate, D., Abate, K. H., et al. (2018). Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the global burden of disease study 2017. *The Lancet*, 392(10159):1789–1858.

Lech, M., Low, L.-S., and Ooi, K. E. (2014). Detection and prediction of clinical depression. In *Mental Health Informatics*, pages 185–199. Springer.

Li, G., Zhou, X., Lu, T., Yang, J., and Gu, N. (2016). SunForum: Understanding Depression in a Chinese Online Community. In *Proc. of the 19th ACM Conf. on Comput.-Supported Cooperative Work & Social Comput.*, pages 514–525, New York, NY, USA. ACM Press.

Nakagawa, E. Y., Scannavino, K. R. F., Fabbri, S. C. P. F., and Ferrari, F. C. (2017). *Revisão sistemática da literatura em engenharia de software: teoria e prática*. Elsevier Brasil.

Park, S., Kim, I., Lee, S. W., Yoo, J., Jeong, B., and Cha, M. (2015). Manifestation of Depression and Loneliness on Social Networks: A Case Study of Young Adults on Facebook. In *Proc. of the 18th ACM Conf. on Comput. Supported Cooperative Work Social Comput.*, CSCW '15, pages 557–570, New York, NY, USA. ACM.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *J. of Manage. Inf. Syst.*, 24(3):45–77.

Razis, G., Anagnostopoulos, I., and Zeadally, S. (2020). Modeling Influence with Semantics in Social Networks. *ACM Computing Surveys (CSUR)*, 53(1):1–38.

Sousa, D. N. F. (2016). Automatic Research Areas Identification in C&T. Master's thesis, Universidade Federal do Rio de Janeiro.

Tsugawa, S., Kikuchi, Y., Kishino, F., Nakajima, K., Itoh, Y., and Ohsaki, H. (2015). Recognizing Depression from Twitter Activity. *Proc. of the 33rd Annu. ACM Conf. on Human Factors in Computing Systems - CHI '15*, pages 3187–3196.

Vedula, N. and Parthasarathy, S. (2017). Emotional and Linguistic Cues of Depression from Social Media. *Proc. of the 2017 International Conference on Digital Health - DH '17*, pages 127–136.

Wieringa, R. J. (2014). *Design Science Methodology for Inf. Syst. and Softw. Eng.* Springer Berlin Heidelberg, Berlin, Heidelberg.

Musical Genre Analysis Over Dynamic Success-based Networks

Gabriel P. Oliveira[1], **Anisio Lacerda**[1], **Mirella M. Moro**[1]

[1]Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte – MG – Brazil

{gabrielpoliveira,anisio,mirella}@dcc.ufmg.br

Nível: Mestrado
Data de ingresso: Março de 2019
Data prevista para conclusão: Março de 2021
Etapas concluídas: Revisão da literatura;
Definição do problema; Coleta e pré-processamento dados;
Organização e construção do conjunto de dados;
Modelagem e caracterização das redes de colaboração;
Detecção de perfis de colaboração entre gêneros musicais.
Publicações: [Oliveira et al. 2020]

Abstract. *As the music industry becomes more complex, reaching a wider audience through collaboration is effective in maintaining the relevance of artists from distinct genres in the market. As genre is one of the most prominent high-level music descriptors, all music-related analyses may depend on it. In this study, we propose to analyze the relation between musicians teaming up on a hit song with its success under a genre perspective. Our methodology includes building success-based genre collaboration networks to detect collaboration profiles and studying their evolution over time. With this work, we aim to provide potential impact to both the research community and the music industry.*

Resumo. *À medida em que a indústria da música se torna mais complexa, a estratégia de alcançar maiores públicos pela colaboração tem se mostrado efetiva ao manter a relevância de artistas de diferentes gêneros no mercado. Como o gênero é uma das principais características de uma música, todas as análises nesse contexto podem ser dependentes dele. Neste estudo, propõe-se analisar a relação entre a colaboração entre artistas com o sucesso musical sob a perspectiva de gênero. Nossa metodologia inclui a construção de uma rede de gêneros musicais baseadas no sucesso para detectar perfis de colaboração e o estudo de sua evolução através do tempo. Com este trabalho, espera-se proporcionar um potencial impacto para a comunidade acadêmica e a indústria da música.*

1. Introduction

Music is not only one of the world's most important cultural industries, but also one of the most dynamic. Over the last few decades, the world has seen a dramatic change in the way people consume music, moving from physical records to streaming services. Few years ago, songs and their videos needed to be played on the radio and TV to be successful; but today, they can be easily accessed on digital platforms such as Spotify and YouTube. Since 2017, streaming services have become the main source of revenue within the global recorded music market, mainly due to the fans' engagement and adoption of these platforms. In fact, their revenues increased by 75.4% from then, reaching US$ 11.4 billion by the end of 2019[1]. As a result, artists are encouraged to reinvent strategies to maintain their presence in the market and reach new audiences.

As the music industry becomes more complex and competitive, artist collaboration has grown into one of the main strategies to promote new songs and acquire new audience. This widely adopted strategy is a strong force driving music nowadays, maintaining artists' relevance in the market. Such connections usually help artists bridge the gap between styles and genres, overlapping new fan bases and consequently increasing their numbers. In such a way, several studies approach the factors behind musical success, creating an emerging field within computer science called Hit Song Science (HSS). Collaboration-aware studies then become promising, as successful artists are more likely to have a high degree of collaboration in success-based networks [Silva et al. 2019]. In fact, there is strong evidence in the literature that factors leading to an ideal musical partnership can be understood by exploring collaboration patterns that directly impact its success [Bryan and Wang 2011].

The genre perspective is very important when analyzing the impact of collaborations in musical success, as each genre has a distinct audience that behaves in its own way. Figure 1 shows this phenomenon and highlights the growing trend in the number of collaborations within Billboard Hot 100 Charts. Although the general curve increases over time, genres such as *pop* and *R&B* present a collaboration rate higher than others (e.g., *rock*). This contrast can be explained by the intrinsic nature of each music genre. For instance, *pop* and *R&B* artists frequently collaborate with the *rap* community, mainly as featured artists. Also, partnerships involving *pop* music may take place not only through intra-genre collaborations but also through inter-genres, bringing an additional dimension to their songs. For example, in April 2019, the collaboration between the American pop singer Halsey and the k-pop group BTS in the song *Boy With Luv* became the most viewed YouTube music video in 24 hours and reached #8 on Billboard Hot 100 Chart. As this creative market changes, it becomes more unpredictable; and doing both predictive and diagnostic analyses in such a context remains challenging.

This work aims to better understand the dynamics of the music industry, specifically the relation between artist collaboration and musical success under the genre perspective. Although building and studying success-based artist networks are already subject of our group recent research [Silva et al. 2019, Silva and Moro 2019], to the best of our knowledge, there are no studies considering how the artist's genre may influence the popularity of a song. For example, in the past few years the collaborations between *pop* and *reggaeton* artists have become more frequent and sucessful, mostly due to the stardom

[1] IFPI Global Music Report 2019. https://gmr.ifpi.org/

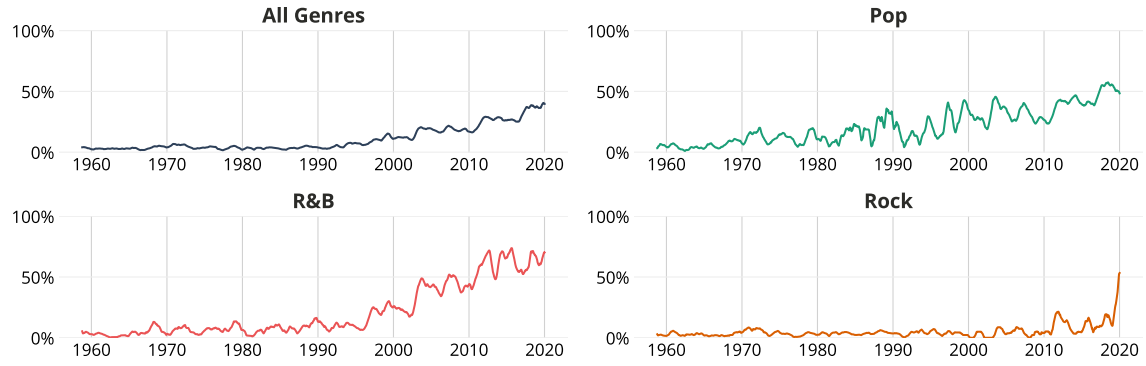


Figure 1. Historical frequency of collaborative hit songs for selected genres on Billboard Hot 100 Chart (1958 - 2020).

of the hit *Despacito* by Luis Fonsi and Daddy Yankee in 2017. This song gained a remix with the Canadian pop singer Justin Bieber, reaching the top of Billboard Hot 100[2] for 16 consecutive weeks.

Motivation and Relevance. Musicians teaming up is nothing new but has risen far beyond the norm. Remaining an industry of creative growth, it is only natural for music (i.e., all musical scene members) adapting to new conditions and redefining its layout. Not surprisingly, the Grammy[3] categories were tightened (from 109 to 78, in 2012) as a result of music's dynamic nature. That is, the notion of categories and genres are blurred as never before. Through cross-genre collaboration, artists are venturing into new domains and working outside of the category which they had originally been ascribed to. Such a collaboration phenomenon may be drastically reshaping music global environment, by challenging segments of certain genres to come up with something entirely new.

This novel and dynamic environment brings high volumes of data about songs, their characteristics, and the social interactions about them. The popularization of digital platforms allows people all over the world to have access and interact with content in real-time [Barbosa et al. 2013, Harb and Becker 2018], increasing the cultural connection between distinct parts of the globe, while each market maintains its unique characteristics. Therefore, this work provides relevant contributions to the Database field by collecting, aggregating, modeling, and analyzing data obtained from different sources on the Web, in addition to processing and enriching social data (e.g., the collaboration network between artists and genres). We also aim to organize and provide a unique dataset on musical success focusing on genre collaboration, with information from charts, songs, and artists.

Research Goals. As the collaboration phenomenon becomes stronger over the years, it is necessary to explore all factors that make it so relevant nowadays. Therefore, this work aims to analyze artist collaboration under a genre perspective to better understand how the genre connections impact musical success. Specifically, we plan to:

RG1. Build a proper musical success dataset with enhanced genre collaboration data;
RG2. Model a success-based genre collaboration network considering distinct regional markets, as well as the global aggregated scenario;

[2]The Billboard Hot 100 is the main weekly song chart within the United States. A song's position in the chart is calculated by considering sales, radio plays and streaming count.

[3]Grammy Awards: `https://en.wikipedia.org/wiki/Grammy_Award`

RG3. Detect and evaluate the collaboration profiles within the genre network;
RG4. Evaluate the dynamics of both the network and the profiles over time, as well as their relation with musical success.

2. Related Work

Genre is fundamental within the musical scenario by aggregating songs that share common features. Hence, it is often used in the field of Music Information Retrieval (MIR), which aims to extract relevant information from music content. Indeed, several tasks are genre-dependent or directly related to them, such as automatic genre classification, which has been largely studied by the MIR community [Ghosal and Sarkar 2020]. Nonetheless, there are also genre-aware studies assessing genre modeling [Prockup et al. 2015], preferences [Bansal and Woolhouse 2015], disambiguation/translation [Hennequin et al. 2018, Epure et al. 2019], new datasets [Bogdanov et al. 2019], and ontologies [Schreiber 2016]. Network science, the core of our methodology, has also been used to model genres into influence networks [Bryan and Wang 2011] and song communities [Corrêa et al. 2011].

Hit Song Science (HSS) tackles the problem of predicting the popularity of a given song, and is also an emerging field within MIR. Thus, different studies analyze the impact of acoustic and social features in musical success. In the early years of HSS, only acoustic features (i.e., the internal technical aspects of a song, such as timbre, mode and key) were assessed by researchers [Dhanaraj and Logan 2005]. Nonetheless, as the Web became popular and widely adopted, social interactions were included as features in prediction models. For instance, Cosimato et al. [2019] predict an album success through users' interactions on social networks such as Twitter, Instagram and YouTube. Other studies include genre information in their models [Zangerle et al. 2019], although its impact on success is not deeply evaluated.

Moreover, Silva et al. [2019] address collaboration as a key factor in success, using topological properties to detect relevant profiles in artist networks. In a later study, the causality between collaboration and success is addressed [Silva and Moro 2019], increasing the knowledge and reinforcing the relevance of the collaboration phenomenon in the musical scenario. In fact, such an approach is novel and promising in HSS, but it is restricted to the artist and song levels. In addition, these and most of the aforementioned studies regarding musical success only consider data from American charts, mainly Billboard Hot 100. This may be due to the ease of obtaining data but it may not reflect the whole global scenario, as each country has its own distinct behavior when consuming music, which includes preferred artists and genres.

Contributions. Studying collaboration from a genre perspective may reveal important information on how artists from different communities team up to make a new hit song. To the best of our knowledge, we are the first to build a success-based genre network, investigating its evolution over time and the collaboration profiles within it, going deeper into the potential intrinsic factors that make up a successful collaboration. Likewise, the approach considering regional markets makes this work more realistic, as local engagement shapes the global environment. We combine a precise heterogeneous data collection with proper modeling to enhance further data analysis by scientists and record labels CEOs. Therefore, this work sheds light on the science behind the collaboration phenomenon, providing potential impact to both the Databases community and the music industry.

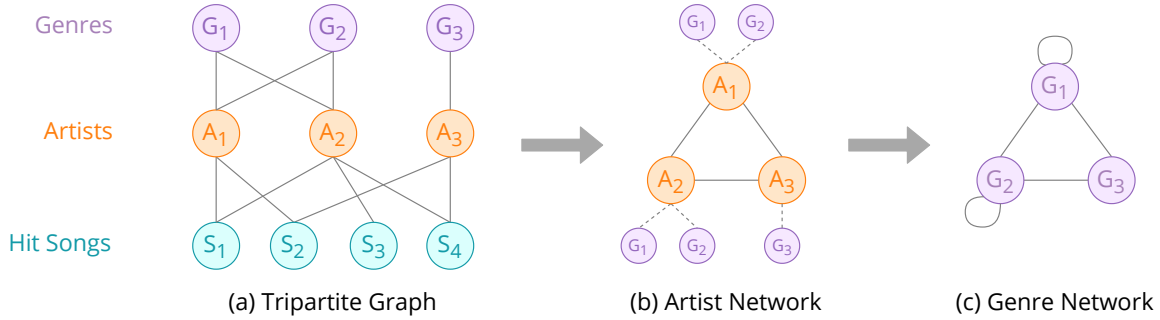


Figure 2. Reduction from the tripartite (a) to the one-mode Genre Collaboration Network (c). The intermediate step is an Artist Network with genre information (b). Artists and genres are linked when hit songs involve both nodes.

3. Methodology

In this section, we focus on the main steps of the proposed methodology to assess our research goals. These steps include data collection, the collaboration network modeling, plus the metrics and techniques that can be used in our experiments and evaluation.

Data Collection and Processing. Since 2017, streaming services have become the main source of revenue within the global recorded music market, with an increase in their revenues of 75.4% from then. Thus, we obtain our data from Spotify, the most popular global audio streaming service, with more than 286 million users across 79 markets[4]. It provides a weekly chart of the 200 most streamed songs in all its markets, and an aggregated global chart. We collect global and regional charts as from 2017, considering eight of the top 10 music markets[5] according to IFPI: United States, Japan, United Kingdom, Germany, France, Canada, Australia, and Brazil. We also use Spotify API to gather information about the hit songs and artists within the charts, such as all collaborating artists within a song (since the charts only provide the main ones) and their respective genres, which is the core of this work.

Genre Network Modeling. A Collaboration Network is usually modeled as a graph formed by nodes (vertices) that may be connected through edges. For analyzing the interactions between genres, we model music collaboration as a tripartite graph, in which nodes are divided into three sets: genres, artists, and hit songs. The building process of the genre network from the tripartite model is illustrated in Figure 2. Collaborative hit songs are sung by two or more artists, regardless of their participation (e.g., a *feat.* or a duet). We also equally consider all genres linked to an artist because they shape how such an artist is seen by fans and music industry. We then reduce the tripartite model into a one-mode network in which nodes are exclusively genres. However, such a reduction is only possible by executing an intermediate step: building the artist collaboration network, Figure 2(b). In such a network, two artists are connected when both collaborate in one or more hit songs. The genres are not lost, as they are linked directly to the artists. We may now build the final network by connecting the genres of artists who collaborate in the artist network. The edges are undirected and weighted by the number of hit songs involving artists from both genres, Figure 2(c). Also, self-loop edges are allowed, as there are hit songs from artists of the same genre.

[4]Spotify Company Info: `https://newsroom.spotify.com/company-info/`
[5]Data from South Korea and China was not available in Spotify.

Profiling and Evaluation. After building the success-based genre collaboration networks, it is necessary to characterize them considering each distinct market and their evolution over time. We do so by analyzing network science metrics such as degree and weighted degree, clustering coefficient, and density. Next, we plan to use a combined approach of network science metrics and clustering algorithms to detect collaboration profiles within music genres and then investigate their relation with musical success. Distinguishing cross-genre collaborations from intra-genre ones is fundamental for our analyses, as we believe that crossing genre frontiers may bring more success for a song, as it will join distinct but powerful audiences to leverage the song's numbers. Finally, we intend to use the collaboration profiles and other genre features in more specific tasks such as collaboration prediction and recommendation.

4. Preliminary Results

The collaboration network characterization is the initial step of our evaluation, and we analyze global and each market separately. The global genre networks reveal the world is more open to new successful genres (number of nodes/genres growth). Also, the degree analysis indicates that low-degree emerging genres may become popular shortly, expanding their collaborations to other unexplored genres. For instance, *k-pop* connections double as it spreads worldwide, approaching genres such as *reggaeton* (e.g., the collaboration between J-Hope from BTS and Becky G in the song *Chicken Noodle Soup*, September 2019). For regional markets, we classify the countries into three groups, according to the similarities in networks' evolution: *(i)* USA and Canada; *(ii)* Brazil, France, Germany and Japan; *(iii)* UK and Australia. Overall, considering regional markets individually becomes more important for producers and record labels, as they are delivering more global hits over time. Their distinct behavior emphasizes the strength of cultural aspects on determining how music is consumed and the success of a given genre or artist.

Next, for each country and year, we detect four distinct clusters within the genre networks and investigate the relation between these groups and musical success (i.e. hit songs present in Spotify charts, evaluated by their amount of streams). In short, the collaboration profiles discovered are: *(i) Solid*, composed of well-established collaborations between most popular genres (super-genres), which have been going on for decades; *(ii) Regular*, composed of the most common collaborations in all markets, which are very similar to solid collaborations but not as engaged; *(iii) Bridge*, composed of collaborations with high influence, representing bridge-like connectors between two regions of a network (mostly between divergent music styles); and *(iv) Emerging*, formed mainly of collaborations between regional genres. Such partnerships generally occur within the same genre. Hence, detecting such profiles is a powerful way to assess musical success by describing similar behaviors within collaborative songs from multiple angles.

These and other preliminary results are present in a paper recently accepted for publication in the 21st International Society for Music Information Retrieval Conference (ISMIR) [Oliveira et al. 2020]. As future work, we plan to continue investigating the evolution of genre collaborations, specifically the dynamics of the collaboration profiles. We also aim to address other open research issues regarding the science behind musical success (e.g., recommending collaborations) by using different data mining and machine learning techniques to extract meaningful knowledge about the music domain, such as mining frequent genre patterns within successful collaborations.

References

Bansal, J. and Woolhouse, M. (2015). Predictive power of personality on music-genre exclusivity. In *ISMIR*, pages 652–658.

Barbosa, G. A. R., Holanda, P. H. F., dos Santos, G. E., da Costa, C. C., Silva, I. S., Veloso, A., and Meira Jr., W. (2013). Caracterização do uso de hashtags do twitter para mensurar o sentimento da população online: Um estudo de caso nas eleições presidenciais dos EUA em 2012. In *SBBD (Short Papers)*, pages 19:1–19:6. SBC.

Bogdanov, D. et al. (2019). The acousticbrainz genre dataset: Multi-source, multi-level, multi-label, and large-scale. In *ISMIR*, pages 360–367.

Bryan, N. J. and Wang, G. (2011). Musical influence network analysis and rank of sample-based music. In *ISMIR*, pages 329–334.

Corrêa, D. C., Levada, A. L. M., and da F. Costa, L. (2011). Finding community structure in music genres networks. In *ISMIR*, pages 447–452.

Cosimato, A. et al. (2019). The conundrum of success in music: Playing it or talking about it? *IEEE Access*, 7:123289–123298.

Dhanaraj, R. and Logan, B. (2005). Automatic prediction of hit songs. In *ISMIR*, pages 488–491.

Epure, E. V., Khlif, A., and Hennequin, R. (2019). Leveraging knowledge bases and parallel annotations for music genre translation. In *ISMIR*, pages 839–846.

Ghosal, S. S. and Sarkar, I. (2020). Novel approach to music genre classification using clustering augmented learning method (CALM). In *AAAI MAKE*, volume 2600 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Harb, J. G. D. and Becker, K. (2018). Emotion analysis of reaction to terrorism on twitter. In *SBBD*, pages 97–108. SBC.

Hennequin, R., Royo-Letelier, J., and Moussallam, M. (2018). Audio based disambiguation of music genre tags. In *ISMIR*, pages 645–652.

Oliveira, G. P., Silva, M. O., Seufitelli, D. B., Lacerda, A., and Moro, M. M. (2020). Detecting collaboration profiles in success-based music genre networks. In *ISMIR*.

Prockup, M. et al. (2015). Modeling genre with the music genome project: Comparing human-labeled attributes and audio features. In *ISMIR*, pages 31–37.

Schreiber, H. (2016). Genre ontology learning: Comparing curated with crowd-sourced ontologies. In *ISMIR*, pages 400–406.

Silva, M. O. and Moro, M. M. (2019). Causality analysis between collaboration profiles and musical success. In *WebMedia*, pages 369–376. ACM.

Silva, M. O., Rocha, L. M., and Moro, M. M. (2019). Collaboration Profiles and Their Impact on Musical Success. In *ACM/SIGAPP SAC*, pages 2070–2077, Limassol, Cyprus.

Zangerle, E. et al. (2019). Hit song prediction: Leveraging low- and high-level audio features. In *ISMIR*, pages 319–326.

Usando relacionamentos entre atributos no processo de Fusão de Dados

Gabrielle Karine Canalle[1], **Ana Carolina Salgado**[1], **Bernadette Farias Lóscio**[1]

[1]Centro de Informática - Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Recife– PE – Brazil

{gkc, acs, bfl}@cin.ufpe.br

Nível: Doutorado

Mês e Ano de Ingresso: março/2017

Mês e Ano Previstos para Defesa: fevereiro/2021

Etapas Concluídas: Créditos em Disciplinas, Exame de Qualificação, Definição do Problema, Especificação da Solução e da Metodologia de Avaliação, Levantamento Bibliográfico.

Etapas Futuras: Implementação da solução, Realização da Avaliação com Experimentos e Finalização da Escrita da Tese.

Publicações: *A Survey on Data Fusion: What for? In what form? What is next? - Journal of Intelligent Information Systems*
(Aceito para publicação, em processo final de revisão)

Resumo. *A Fusão de Dados é uma tarefa primordial quando se deseja integrar dados, e avaliar a qualidade das fontes no processo de Fusão de Dados tem se tornado imprescindível. No entanto, em cenários de dados na Web, onde frequentemente ocorre o fenômeno Long-tail, é difícil avaliar a confiabilidade das fontes de forma precisa. Neste sentido, este trabalho propõe uma abordagem para descoberta de relacionamentos entre atributos, que serão utilizados para inserir conhecimento adicional no processo de Fusão de Dados. Deste modo, a avaliação de confiabilidade das fontes será realizada de forma mais eficiente, não apenas com base nos dados de entrada, mas também no conhecimento adicional extraído dos relacionamentos.*

1. Introdução e Motivação

Nos últimos anos a quantidade de dados e fontes de dados na *Web* têm aumentado continuamente, em domínios variados. Os dados podem ser de sensores, mídias sociais, diferentes *websites*, ou ainda empresas privadas. Uma grande quantidade desses dados contêm valores errôneos, ausentes e conflitantes. Simultaneamente ao crescimento de informações disponíveis, o consumo de dados também cresce.

Os dados são consumidos por usuários ou por aplicativos, que podem ser afetados por informações errôneas, levando ambos a tomarem decisões incorretas. Por isso, a área de descoberta da verdade (*Truth Discovery*) tem recebido crescente atenção. As soluções, geralmente não supervisionadas, têm como ideia principal que valores verdadeiros são fornecidos por fontes de dados confiáveis, e fontes de dados são confiáveis se fornecem valores verdadeiros. O objetivo é resolver os conflitos existentes nos dados, identificando os valores corretos e abordando a veracidade dos dados durante o processo. [Berti-Équille and Borge-Holthoefer 2015] As técnicas atuais geralmente são iterativas em duas etapas até a convergência: i) avaliar a confiabilidade das fontes, ii) avaliar a confiança dos valores.

Ao longo dos últimos anos, foram propostas diversas soluções que utilizam a confiabilidade das fontes na descoberta da verdade [Li et al. 2017] [Broelemann and Kasneci 2018], [Zhang et al. 2018]. Entretanto, em cenários em que ocorre o fenômeno *Long-tail*, onde a maioria das fontes fornece valores para apenas alguns atributos de entidades, enquanto apenas algumas fontes cobrem vários atributos, avaliar a confiabilidade das fontes e utilizá-la no processo de resolução de conflitos pode ser insuficiente [Broelemann and Kasneci 2018]. A eficácia da estimativa de confiabilidade das fontes é fortemente afetada pelo número total de informações providas por cada fonte. Quando uma fonte de dados provê poucos dados, como a maioria das fontes nos cenários *Long-tail*, se torna um desafio estimar sua confiabilidade precisamente. Neste trabalho, estamos interessados na Fusão de Dados em cenários de dados na *Web*, onde geralmente ocorre o fenômeno *Long-tail*.

2. Caracterização da Contribuição

Dado um conjunto de Fontes de Dados que provê valores para atributos de uma dada entidade, assumindo que o processo de Resolução de Entidades já foi realizado [Vieira et al. 2019], como podemos identificar os valores corretos para o conjunto de atributos de representações de uma dada entidade?

Para isso, propomos descobrir relacionamentos entre atributos e utilizá-los no processo de Fusão de Dados. Podem existir diferentes relacionamentos entre atributos, por exemplo, um atributo $cidade = \text{“}Recife\text{”}$ está relacionado a outro atributo $estado = \text{“}Pernambuco\text{”}$, ou ainda, um atributo $cep = \text{“}50740 - 132\text{”}$ é relacionado a um atributo $cidade = \text{“}Recife\text{”}$. Propomos capturar três tipos de relacionamentos: hierárquicos, de composição e dependência funcional aproximada (DFA).

Uma dependência funcional (DF) expressa um relacionamento entre atributos de um conjunto de dados. No entanto, neste trabalho seguiremos o conceito de DFA [Huhtala et al. 1999], [Mandros et al. 2017], [Kruse and Naumann 2018], que relaxa a definição de dependência funcional, permitindo que algumas representações de entidade

violem a DF. Além disso, como o conjunto de representações de entidade é proveniente de diversas fontes e podem existir valores faltantes, pretendemos adaptar a descoberta de dependências aproximadas seguindo a solução proposta em [Berti-Équille et al. 2018].

Finalmente, a partir do conhecimento extraído dos relacionamentos, acreditamos que o processo de descoberta da verdade pode se tornar mais eficaz, já que o cálculo de confiabilidade da fonte e confiança do valor serão realizados não apenas com base nos dados de entrada, mas também no conhecimento adicional obtido. Deste modo podemos auxiliar no processo de descoberta da verdade penalizando ou reforçando os valores de confiabilidade das fontes e confiança dos valores, com base nesse conhecimento adicional.

2.1. Hipótese

Utilizar relacionamentos entre atributos pode auxiliar no processo de descoberta da verdade e melhorar a eficácia da Fusão de Dados, principalmente em cenários de *Long-tail*.

2.2. Definições Preliminares

Nesta seção, apresentamos algumas notações e definições de conceitos fundamentais utilizados para a especificação da estratégia proposta.

Fonte de Dados. Uma Fonte de Dados f fornece dados sobre entidades do mundo real. Um conjunto de Fontes de Dados é representado por $F = \{f_1, f_2, ..., f_n\}$.

Entidade. Uma entidade e é um conceito do mundo real, como uma música, um livro, ou uma pessoa. Cada entidade é expressa por um conjunto de atributos $A = \{a_1, a_2, ...a_m\}$.

Representação de Entidade. Uma representação de entidade r_n é a representação de uma entidade e_j, fornecida por uma fonte de dados f_i, denotada por $f_i.r_n$. Uma representação de entidade r_n é definida por um conjunto de pares $\{(a_1, v_1), (a_2, v_2), ..., (a_j, v_k)\}$, tal que $a_i \in A$, e v_i é o valor de a_i para a entidade e_j na fonte f_i.

Conjunto de Representações de Entidade. Seja $R = \{(r_1, f_1), (r_2, f_2), ...(r_i, f_k)\}$ um conjunto de representações de entidade oriundo de diferentes fontes de dados, onde todas as representações de entidade estão relacionadas a uma mesma entidade e do mundo real.

Dependência Funcional. Uma dependência funcional DF se dá quando os valores de um atributo a_x determinam os valores de um atributo a_y, e é expressa como DF: $a_x \rightarrow a_y$.

Dependência Funcional Aproximada. Uma dependência funcional aproximada DFA relaxa parcialmente o conceito de Dependência Funcional, e requer que a DF seja satisfeita pela maioria das representações de entidade, ou seja, permite que algumas representações de entidade violem a DF [Huhtala et al. 1999].

3. Solução Proposta

Propomos uma solução baseada em inserir conhecimento adicional no processo de Fusão de Dados por meio da descoberta de relacionamentos entre atributos. Pretendemos deixar claro que não estamos interessados em propor uma solução nova para Fusão de Dados, e sim uma abordagem adaptável que se propõe a melhorar os resultados dos algoritmos do estado da arte. A Figura 1 apresenta a arquitetura da abordagem proposta, composta por dois módulos: i) descoberta de relacionamentos; ii) de descoberta da verdade.

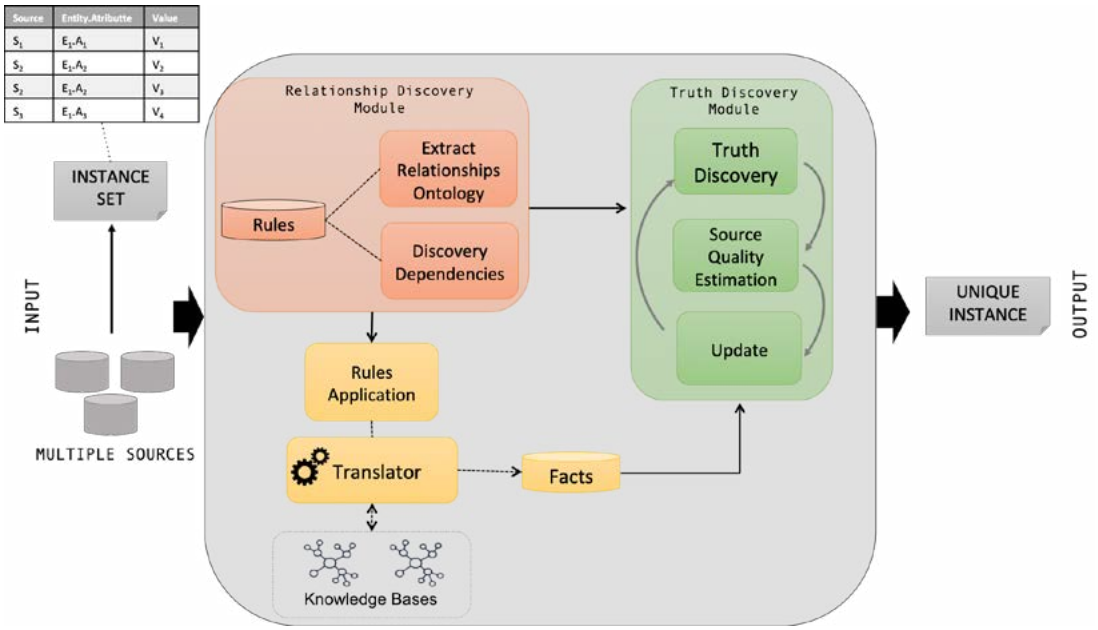


Figura 1. Arquitetura da abordagem proposta.

Módulo de Descoberta de Relacionamentos Neste módulo, os relacionamentos entre atributos são descobertos, e armazenados na forma de regras em um repositório. As regras são aplicadas sobre os dados, gerando afirmações. Essas afirmações passam pelo tradutor e são transformadas em consultas SPARQL para posterior busca na *Web*. Quando uma afirmação após a busca é dada como verdadeira, dizemos que a afirmação é válida, e ela então é armazenada no repositório de fatos. Caso contrário, a afirmação é desconsiderada.

Módulo de Descoberta da Verdade - No módulo de Descoberta da Verdade, os modelos existentes podem ser implementados/adaptados [Fang 2017, Broelemann and Kasneci 2018, Zhang et al. 2018]. Em sua maioria, esses modelos são iterativos em duas etapas: avaliação da confiança do valor e avaliação de confiabilidade das fontes. Após o módulo de descoberta de relacionamentos ser executado e as consultas na *Web* obterem as afirmações corretas, o conhecimento armazenado no repositório de fatos é utilizado para reforçar ou penalizar os valores de confiança e confiabilidade.

3.1. Descrição do Processo

Para extrair relacionamentos hierárquicos e de composição são utilizadas ontologias, já que elas são empregadas para representar o conhecimento de maneira formal e reutilizável. Deste modo, é possível extrair conhecimento a priori sobre relacionamentos entre atributos, como relacionamentos *isPartOf* e *isA*, e utilizá-los no processo de Descoberta da Verdade. Relacionamentos do tipo DFA são encontrados por meio de algoritmos de descoberta de DFA's [Huhtala et al. 1999, Kruse and Naumann 2018, Mandros et al. 2017].

O processo de descoberta de relacionamentos está ilustrado no Algoritmo 1, e recebe como entrada um conjunto de representações de uma dada entidade oriundas de múltiplas fontes. A partir do conjunto de dados de entrada, é criada uma representação completa em que todas as representações de entidade são unificadas em uma tabela. As colunas são todos os atributos fornecidos pelas fontes. A seguir, é realizada a descoberta de relacionamentos entre atributos. Os relacionamentos são utilizados para gerar um con-

junto de regras (ex: $R_i = atributo_j \rightarrow atributo_k$) que serão aplicadas nos dados para extrair afirmações (ex: $A_1 = valor\_atributo_j \rightarrow valor\_atributo_k$). As afirmações serão traduzidas em consultas, e buscadas na *Web*. As afirmações válidas são dadas como fatos e armazenados por domínio no repositório de fatos para posterior utilização.

Para um mesmo domínio, se existir outro conjunto de dados como entrada, os repositórios de regras e fatos estarão disponíveis para consulta, sendo necessário realizar a descoberta de relacionamentos apenas para os atributos que não passaram pelo processo anteriormente. Quanto mais o processo é realizado, mais aumentam as informações no repositório e o conhecimento prévio sobre os dados, o que tende a melhorar o processo.

Algoritmo 1 Algoritmo para descoberta de relacionamentos

1: **function** DESCOBRERELACIONAMENTOS(R, A, $Regras$)
▷ R corresponde ao conjunto de representações da entidade e_i;
▷ A corresponde ao conjunto de atributos de e_i;
2: $AFIRM' \leftarrow \emptyset$ ▷ inicializa o conjunto de novos relacionamentos
3: $Q \leftarrow \emptyset$ ▷ inicializa query de busca
4: $Rel \leftarrow \emptyset$ ▷ inicializa o conjunto de relacionamentos
5: $Rel_{temp} \leftarrow \emptyset$ ▷ inicializa o repositório temporário
6: **for all** $a_i \in A$ **do** ▷ Percorre cada atributo a_i do conjunto de atributos A
7: $Rel \leftarrow verificaRegras(a_i, Regras)$ ▷ verifica se já existem regras para o atributo a_i no repositório de regras
8: **if** $Rel \neq \emptyset$ **then**
9: $Rel_{temp} \leftarrow verificaRegras(a_i)$ ▷ carrega as regras existentes para a_i no repositório temporário
10: **else**
11: $Rel \leftarrow descobreRelacionamentos(a_i)$ ▷ descobre os relacionamentos para os atributo a_i
12: $Rel_{temp} \leftarrow Rel$
13: $Regras \leftarrow armazenaRepositorioRegras(Rel)$ ▷ armazena os novos relacionamentos descobertos no repositório de regras e no repositório temporário
14: **end if**
15: $AFIRM \leftarrow aplicaRegras(Rel_{temp})$ ▷ aplica as regras do atributo a_i nos dados e gera as afirmações
16: $Q \leftarrow traduzAfirmacoes(AFIRM)$ ▷ traduz as afirmações em linguagem de consulta
17: $AFIRM' \leftarrow realizaBusca(Q)$ ▷ realiza a busca das afirmações na *Web* e retorna um subconjunto de afirmações válidas
18: $Fatos \leftarrow armazenaRepositorioFatos(AFIRM', a_i)$
19: **end for**
20: **end function**

Na descoberta da verdade pode-se adaptar qualquer solução do estado da arte [Fang 2017, Broelemann and Kasneci 2018, Zhang et al. 2018]. Na etapa de avaliação de confiança dos valores que é feita para cada atributo do conjunto de dados, verifica-se no repositório de fatos se existe algum fato cujo atributo a ser identificada a verdade se encontre ao lado direito do fato. Se não existir, o processo de descoberta da verdade para

este atributo é realizado normalmente. Se existir, significa que existem fatos que podem ajudar no processo. Os fatos são utilizados para ajudar a calcular a confiança do valor e a confiabilidade das fontes, reforçando ou penalizando esses valores. A saída do processo é uma representação de entidade única o mais completa possível. Se desejável, o valor de qualidade das fontes também pode ser retornado. Mais detalhes sobre esta etapa e o algoritmo que ilustra o processo podem ser encontrados no documento de qualificação.

4. Trabalhos Relacionados

Existem poucos trabalhos na literatura que se propõem a lidar com relacionamentos entre atributos. Dentro de nosso conhecimento, podemos citar [Pasternack and Roth 2010], [Nakhaei and Ahmadi 2017], [Pradhan et al. 2018] e [Beretta et al. 2018]. No nosso trabalho, além de abordar relacionamentos a partir de ontologias, como em [Pradhan et al. 2018] e [Beretta et al. 2018], pretendemos utilizar DFA's para encontrar outros relacionamentos existentes entre atributos. O que também se difere é que queremos aplicar os relacionamentos de maneira diferente dos trabalhos citados. Em nosso trabalho, construiremos uma base de fatos a partir de consultas realizadas na *Web*. Os relacionamentos são utilizados para gerar essas consultas. Não estamos interessados em propor uma solução nova para Fusão de Dados, nosso objetivo é que nossa solução possa ser utilizada para melhorar algoritmos de descoberta da verdade já existentes.

5. Avaliação dos Resultados

Experimento 1. A partir do levantamento dos algoritmos do estado da arte de descoberta de dependência funcional e suas extensões, tais como dependência funcional aproximada, e dependência funcional condicional, será realizado um experimento para analisar o comportamento desses algoritmos no cenário abordado nesta pesquisa. Para a Fusão de Dados, acreditamos que a dependência funcional aproximada é mais indicada, pois os conflitos existentes nos dados podem violar facilmente as dependências funcionais. Deste modo, descobrir dependências funcionais em dados conflitantes pode não ser muito eficaz. Essa intuição será validada por meio de experimentos. Pretendemos avaliar os algoritmos principalmente em cenários de dados *Long-tail*. O algoritmo identificado como mais apropriado para o nosso contexto será selecionado para utilização e/ou possíveis adaptações.

Experimento 2. Em um segundo experimento, pretendemos analisar qual modelo de descoberta da verdade é mais adequado para ser utilizado/adaptado para o nosso problema. Realizamos um levantamento da literatura e foram identificados quatro trabalhos de descoberta da verdade que se propõem a lidar com cenários *Long-tail* [Broelemann and Kasneci 2018], [Zhang et al. 2018], [Fang 2017]. Pretendemos realizar um comparativo entre esses trabalhos em diferentes cenários, e a partir da análise dos resultados identificar o mais adequado para ser adaptado a nossa solução.

Experimento 3. A partir dos resultados dos experimentos 1 e 2, e com a implementação do protótipo da solução proposta, pretendemos avaliar se nossa solução melhora o resultado dos algoritmos do estado da arte que lidam com cenários *Long-tail*. Iremos realizar um comparativo dos resultados por meio de métricas de qualidade, avaliando os resultados do modelo de Fusão de Dados sem a nossa solução, e utilizando a solução proposta neste trabalho. Com base neste experimento, espera-se investigar as seguintes questões: **Q1.** A solução proposta contribui para melhorar os resultados da Fusão de Dados?; **Q2.** Qual o impacto nos resultados da Fusão de Dados, utilizando a solução proposta?

6. Estado atual do Trabalho

Na fase inicial da pesquisa, realizamos um levantamento exaustivo da literatura, que resultou em um *survey* aceito para publicação no *Journal of Intelligent Information Systems* e está na fase final do processo de revisão. Os experimentos 1 e 2 estão em andamento.

Referências

Beretta, V., Harispe, S., Ranwez, S., and Mougenot, I. (2018). Truth selection for truth discovery models exploiting ordering relationship among values. *K.-B. S.*, 159:29 308.

Berti-Équille, L. and Borge-Holthoefer, J. (2015). *Veracity of Data: From Truth Discovery Computation Algorithms to Models of Misinformation Dynamics*. Synthes Lectures on Data Management. Morgan Claypool Publishers.

Berti-Équille, L., Harmouch, H., Naumann, F., Novelli, N., and Thirumuruganathan, S. (2018). Discovery of genuine functional dependencies from relational data with missing values. *PVLDB*, 11(8):880–892.

Broelemann, K. and Kasneci, G. (2018). Combining restricted boltzmann machines with neural networks for latent truth discovery. *CoRR*, abs/1807.10680.

Fang, X. S. (2017). Truth discovery from conflicting multi-valued objects. In Barrett, R., Cummings, R., Agichtein, E., and Gabrilovich, E., editors, *WWW (Companic Volume)*, pages 711–715. ACM.

Huhtala, Y., Kärkkäinen, J., and Porkka, Pasi, T. H. (1999). Tane: An efficient algorithm for discovering functional and approximate dependencies. *Comput. J.*, 42(2):100–11

Kruse, S. and Naumann, F. (2018). Efficient discovery of approximate dependencies. *PVLDB*, 11(7):759–772.

Li, F., Dong, X. L., Langen, A., and Li, Y. (2017). Discovering multiple truths with a hybrid model. *CoRR*, abs/1705.04915.

Mandros, P., Boley, M., and Vreeken, J. (2017). Discovering reliable approximate functional dependencies. In *KDD*, pages 355–363. ACM.

Nakhaei, Z. and Ahmadi, A. (2017). Toward high level data fusion for conflict resolution. In *ICMLC*, pages 91–97. IEEE.

Pasternack, J. and Roth, D. (2010). Knowing what to believe (when you already know something). In *COLING*, pages 877–885. Tsinghua University Press.

Pradhan, R., Aref, W. G., and Prabhakar, S. (2018). Leveraging data relationships to resolve conflicts from disparate data sources. In *DEXA (2)*, volume 11030 of *Lectu Notes in Computer Science*, pages 99–115. Springer.

Vieira, P. K. M., Lóscio, B. F., and Salgado, A. C. (2019). Incremental entity resolution process over query results for data integration systems. *JIIS.*, 52(2):451–471.

Zhang, J., Wang, S., Wu, G., and Zhang, L. (2018). A effective truth discovery algorithm with multi-source sparse data. In *ICCS (3)*, volume 10862 of *Lecture Notes Computer Science*, pages 434–442. Springer.

Avaliando a Utilização de Weak Supervision na Etapa de Classificação da Resolução de Entidades

Artur Alves de Farias[1]
Orientador: Carlos Eduardo Santos Pires[1]
Co-orientador: Dimas Cassimiro do Nascimento Filho[2]

[1] Programa de Pós-Graduação em Ciência da Computação
Universidade Federal do Campina Grande (UFCG)

[2]Universidade Federal Rural de Pernambuco (UFRPE)

arturfarias@copin.ufcg.edu.br

cesp@dsc.ufcg.edu.br, dimas.nascimentofilho@ufrpe.br

Nível: Mestrado
Mês e Ano de Ingresso: Fevereiro de 2019
Mês e Ano de Previsto para Defesa: Fevereiro de 2021
Etapas Concluídas: Créditos em Disciplinas, Exame de Qualificação.
Etapas Futuras: Realização dos Experimentos e Escrita da Dissertação.

Resumo. *A Resolução de Entidades visa determinar quais registros de uma ou mais bases de dados remetem à mesma entidade no mundo real, podendo ser considerada uma tarefa de classificação. Diversas abordagens de Aprendizagem de Máquina supervisionada foram propostas para tornar a tarefa mais eficaz. No entanto, estas abordagens requerem a existência de conjuntos de dados rotulados. Neste sentido, Weak Supervision, uma abordagem de Aprendizagem de Máquina semi-supervisionada que permite a criação das bases de treinamento de forma programática, surge como uma opção a ser investigada para minimizar o esforço de obter um conjunto de dados rotulados manualmente. Os resultados experimentais preliminares indicam que esta abordagem é capaz de produzir resultados com eficácia bem próximas às abordagens supervisionadas e que necessitam de intervenção humana.*

1. Introdução

A Resolução de Entidades (RE)[1] é a tarefa de identificar registros duplicados, em uma ou mais bases de dados, que remetem à mesma entidade no mundo real [Christen 2012]. Dentre as principais dificuldades para identificar entidades duplicadas, destacam-se a inexistência de identificadores únicos nas entidades, baixa qualidade dos dados e complexidade computacional quadrática das operações [Köpcke et al. 2010].

A tarefa de RE pode ser dividida em quatro etapas: pré-processamento, indexação, comparação e classificação. O pré-processamento realiza o tratamento dos dados de entrada. Na etapa de indexação, o objetivo é diminuir o número de comparações a serem realizadas, desconsiderando os pares de registros que claramente não são duplicados. O resultado desta etapa é um conjunto de pares de registros candidatos a serem comparados. Na etapa de comparação, funções de similaridade são aplicadas nos atributos dos pares de registros candidatos, resultando em um valor de similaridade (ou um vetor de valores de similaridade) para cada par de registros. Por fim, um classificador faz uso desses valores para categorizar os pares de registros em duplicados ou não duplicados. Esta pesquisa foca exclusivamente na etapa de classificação.

Dentre as principais técnicas de classificação utilizadas na tarefa de RE, destacam-se [Christen 2012]: a) *classificação baseada em limiar*: um limite inferior (limiar) é definido e os pares de registros cujo valor de similaridade é superior a este limiar são considerados duplicados; b) *baseada em regras*: regras de decisão, baseadas em heurísticas, senso comum ou conhecimento de especialistas do domínio dos dados, são definidas para classificar os pares como duplicados, caso as regras sejam satisfeitas; c) *baseada em Aprendizagem de Máquina (AM)*: algoritmos de AM são usados para treinar modelos de classificação que posteriormente serão aplicados nos pares de registros classificando-os como duplicados ou não; e d) *baseada em crowdsourcing*: requerem a participação de humanos para auxiliar na classificação dos pares de registros.

As técnicas de classificação apresentam desafios quanto a sua aplicação. Em relação à classificação baseada em limiar, um dos desafios é determinar o valor de limiar apropriado. Para a classificação baseada em regras, um desafio é determinar como as regras podem evoluir à medida que novos dados são adicionados de modo a garantir a eficácia da tarefa. Na classificação baseada em *crowdsourcing*, existe a possibilidade de introdução de erros intencionalmente ou não [Christen 2012]. Frente a estes desafios, cada vez mais são propostos trabalhos que empregam AM [Christophides et al. 2019] na etapa de classificação em RE, em especial a aprendizagem supervisionada, a qual utiliza dados de treinamento rotulados para que algoritmos possam aprender como classificar os pares de registros.

No entanto, assim como as demais técnicas, a classificação baseada em AM apresenta desafios quanto a sua aplicação, sendo um destes a carência de bases de dados de treinamento disponíveis. Uma base de treinamento ideal contém conjuntos de dados representativos e rotulados, necessários para a geração dos modelos preditivos. A obtenção de tais conjuntos de dados é impraticável em muitos cenários devido ao esforço para realizar as rotulações manuais, principalmente para bases de dados de larga escala.

[1] A RE possui diversas nomenclaturas na literatura: *Entity Resolution*, *Entity Matching*, *Data Matching* e *Record Linkage*.

Neste sentido, a abordagem de AM denominada *Weak Supervision* [Zhou 2017] surge como uma opção a ser investigada para resolver o desafio de obter bases de dados de treinamento rotulados. Nesta abordagem, funções de rotulação são criadas e utilizadas para definir rótulos para os dados automaticamente. Os rótulos são empregados no treinamento de um modelo de classificação com o entendimento de que, embora possam ser imperfeitos, podem ser usados para criar um modelo de classificação forte. Logo, esta abordagem alivia o esforço de obter um conjunto de dados rotulados manualmente. Ao permitir de forma programática a geração de bases de treinamento, mais dados rotulados podem ser providos para o treinamento dos modelos de classificação.

Esta pesquisa visa investigar a aplicação de *Weak Supervision* na etapa de classificação da tarefa de RE, na tentativa de reduzir o esforço de se obter conjunto de dados rotulados manualmente. Com isso, algumas áreas de conhecimento ainda não exploradas em RE com AM por falta de bases de treinamento, poderão ser pesquisadas e novos conjuntos de dados para a tarefa de RE serão gerados. Além disso, utilizar *Weak Supervision*, recentemente introduzido na área de AM, na área de RE é uma contribuição importante desta pesquisa. Do melhor do nosso conhecimento, esta relação entre RE e *Weak Supervision* não foi explorada pela comunidade científica até o momento, apenas sugerida, demonstrando o pioneirismo desta pesquisa.

2. Trabalhos Relacionados

Inúmeros trabalhos discutem e aplicam técnicas variadas na etapa de classificação da RE. *Whang et al.* [Whang and Garcia-Molina 2010] exploram o uso de regras de classificação e propõem formas de evoluí-las de forma automatizada. *Santos et al.* [dos Santos et al. 2011] buscam automatizar a definição de um valor de limiar ao verificar a eficácia de vários limiares em *clusters* de pares de registros.

Outros trabalhos aplicam a abordagem de Aprendizagem Ativa (AA) para geração de modelos de classificação [Christophides et al. 2019]. Esta abordagem consiste em selecionar um pequeno conjunto de dados não-rotulados, treinar um modelo com este pequeno conjunto e realizar algumas estimativas de classificação para, então, solicitar que um humano (oráculo) realize marcações manuais em alguns destes dados. Com a comparação entre as predições do algoritmo com as do oráculo, o modelo é retreinado aumentando sua qualidade.

No que se refere à abordagem de *Weak Supervision*, o trabalho [Zhou 2017] comprova que a abordagem não tem necessidade de intervenção humana e que é aplicável a todos os tipos de dados. [Ratner et al. 2016] exploram o paradigma de *Data Programming* que permite que as rotulações sejam feitas por funções criadas programaticamente. Baseado neste paradigma, [Ratner et al. 2017] desenvolveram a ferramenta *Snorkel* que permite gerar conjuntos de dados rotulados mediante a aplicação de estratégias de *Weak Supervision*.

Os autores [Dong and Rekatsinas 2018] afirmam que *Weak Supervision* é uma área promissora para RE por prover dados de treinamento baratos e rápidos. O Snorkel é citado por [Christophides et al. 2019] como uma ferramenta que pode ser utilizada para geração de bases de treinamento para RE. Ambos os trabalhos reforçam que esta pesquisa explora uma área promissora e ainda pouco pesquisada.

Por fim, há trabalhos que aplicam abordagens de AM não supervisionadas em

RE. [Jureka et al. 2017] propõem uma abordagem não-supervisionada para a tarefa de RE combinando várias funções de similaridade. Recentemente, [Primpeli et al. 2020] propõem utilizar AA em RE, porém com um conjunto de dados rotulados anteriormente por uma abordagem não supervisionada, semelhante com a proposta apresentada nesta pesquisa.

De maneira geral, é possível verificar como semelhanças entre os trabalhos relacionados e esta pesquisa, o fato dos trabalhos listados buscarem formas de executarem várias etapas da tarefa de RE programaticamente, assim como sugerirem que a aplicação de *Weak Supervision* em RE é uma área promissora. Porém, esta pesquisa diferencia-se dos trabalhos citados ao realmente aplicar *Weak Supervision* em RE, pois é o primeiro a realizar experimentos na etapa de classificação de RE, além de permitir o surgimento de novas bases de dados de treinamento para RE.

3. Metodologia

A pesquisa se divide basicamente em quatro fases. A primeira fase é dedicada à revisão bibliográfica sobre as principais abordagens usadas na etapa de classificação em RE. Na revisão bibliográfica, também são pesquisados trabalhos referentes à *Weak Supervision*.

A fase seguinte consiste em obter conjuntos de dados reais com gabaritos (necessários para cálculo das métricas de eficácia) para realização das experimentações. Foram buscados conjuntos de dados amplamente utilizados nos trabalhos relacionados a RE (e.g. DBLP-ACM, Amazon-Google). Um maior detalhamento destes conjuntos de dados é apresentado na Tabela 1, onde são apresentadas as quantidades de registros presentes no conjunto de dados, a quantidade de pares de registros que são duplicados, a quantidade total de pares de registros a ser verificada e a proporção de pares de registros que realmente são duplicados sobre a quantidade total de pares de registros.

Tabela 1. Detalhamento dos Conjuntos de Dados

Conjunto de Dados	# Registros	# Pares de Registros Duplicados	# Pares de Registros	Proporção de Pares de Registros Duplicados
DBLP-ACM	2.616 / 2.294	2.224	∼6 milhões	0,074%
Amazon-Google	1.363 / 3.226	1.300	∼4,4 milhões	0,03%
Restaurantes	864	113	∼740 mil	1,5%
CORA	1.879	64.578	∼837 mil	3,66%

A terceira fase é a experimentação da solução proposta. Após a revisão bibliográfica, foram analisados trabalhos sobre as técnicas de *Weak Supervision* e, em muitos destes, o *Snorkel* é citado como uma ferramenta que aplica várias destas técnicas, sendo então selecionada para a fase de experimentação. Esta fase está em curso seguindo a solução proposta, conforme detalhada na Seção 4. Especialistas no domínio dos conjuntos de dados da Tabela 1 codificaram as funções de rotulação, as quais foram aplicadas nos conjuntos de dados. Os resultados parciais desta fase são apresentados na Seção 5.

As variáveis independentes manipuladas são as funções de rotulação, os modelos de classificação, tamanho e proporção dos dados na base de treinamento e a ferramenta de RE.

Por fim, serão analisadas as variáveis dependentes como as métricas de eficácia, Precisão, Cobertura e Medida-F e serão comparadas aos resultados de trabalhos com abordagem supervisionada, como [Köpcke et al. 2010].

4. Solução Proposta

Esse trabalho propõe uma abordagem baseada em *Weak Supervision* para gerar conjuntos de dados de treinamento a serem utilizados na tarefa de RE. A geração é feita de forma programática, baseada em heurísticas, a fim de reduzir o esforço manual. Os conjuntos de dados rotulados são obtidos a partir da aplicação da abordagem de *Weak Supervision* em bases de dados não rotulados, e utilizados para treinar modelos de classificação. Estes últimos são aplicados na etapa de classificação da tarefa de RE. A seguir, é mostrado o fluxo de execução da solução.

Inicialmente, um ou mais especialistas de domínio analisam o conjunto de dados (Passo 1 da Figura 1) e codificam funções de rotulação (Passo 2), baseadas em heurísticas, para realizar a tarefa de RE neste conjunto de dados alvo. Por exemplo, uma função de rotulação pode ser definida de maneira que, caso um certo atributo de um par de registros possua similiaridade maior que 85%, este par de registros deve ser classificado como duplicado; caso contrário, não duplicado.

Figura 1. Especialista analisa conjunto de dados e codifica funções de rotulação.

Os pares de registros não rotulados são lidos pela ferramenta Snorkel onde as funções de rotulação são aplicadas, como mostra o Passo 3 da Figura 2. Cada função determina um rótulo para cada par de registros, sendo **1** para indicar que os registros são duplicados, **0** para não-duplicados e **-1** para se abster da votação. Como resultado, é gerada uma matriz de rótulos (Passo 4). Ao aplicar técnicas de *Weak Supervision*, o Snorkel determina um peso para cada função de rotulação e, por fim, define um rótulo único para cada par de registros, tendo como resultado final a base de dados rotulada para treinamento (Passo 5).

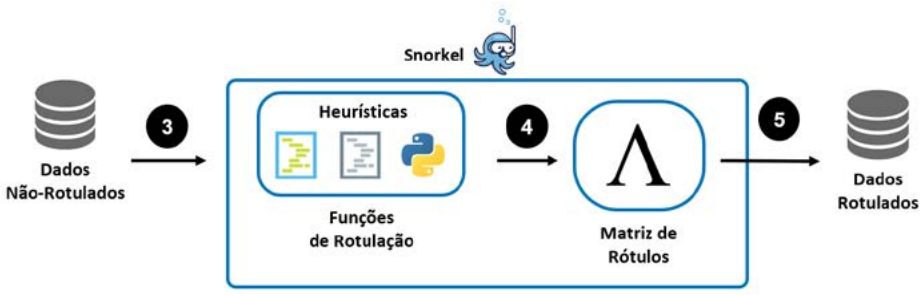


Figura 2. Pares de registros são classificados com *Weak Supervision* e a base de treinamento é gerada.

Com a base de treinamento gerada, um modelo de AM é treinado (Passo 6 da Figura 3) e aplicado na tarefa de RE (Passo 7). Para o passo 7, que consiste na realização

da tarefa de RE, o classificador será acoplado a ferramentas de RE consolidadas para diminuir a ameaça à validade da pesquisa, pois, as etapas anteriores à etapa de classificação não são objetos de estudo desta pesquisa.

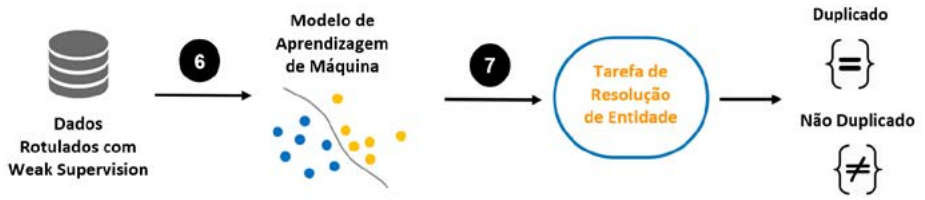


Figura 3. Modelo de classificação é treinado e utilizado na tarefa de RE.

5. Resultados Parciais

Até o momento da escrita deste documento, foram realizados experimentos com três dos quatro conjuntos de dados da Tabela 1, e utilizados cinco modelos de classificação: SVM, Regressão Logística, Árvore de Decisão, Random Forest e AdaBoost. Estes modelos de classificação foram selecionados após análises de *surveys*, entre eles [Köpcke et al. 2010], que comparavam resultados de vários modelos de classificação na área de RE.

Na Figura 4, são apresentados os resultados de Medida-F por modelo de classificação aplicado em cada um dos conjuntos de dados. O trabalho de [Köpcke et al. 2010] também é mostrado para fins de comparação com abordagens supervisionadas que utilizam dados rotulados manualmente para o treinamento dos modelos.

Figura 4. Resultados parciais dos experimentos.

Os resultados preliminares indicam que os modelos de classificação, treinados com as bases de treinamento geradas com *Weak Supervision*, são capazes de alcançar resultados próximos aos melhores resultados das abordagens com AM supervisionada. Por exemplo, para o conjunto de dados DBLP-ACM, obteve-se um valor de Medida-F igual a 0,95 aplicando a solução proposta, um valor de eficácia próximo ao da abordagem supervisionada (0,97). Para o conjunto de dados de Restaurantes, foi obtido como melhor resultado o valor de 0,83, enquanto que a abordagem supervisionada alcançou 0,92. Por fim, para o conjunto de dados Amazon-Google, o melhor resultado foi 0,58, próximo ao 0,62 da abordagem supervisionada.

6. Próximos Passos

Atualmente, a pesquisa encontra-se na fase de experimentação em que experimentos preliminares foram realizados. Pretende-se realizar os experimentos conforme a solução proposta (i.e., com o modelo de classificação treinado embutido em ferramentas de RE),

inclusive, considerando-se utilizar mais conjuntos de dados. Posteriormente, pretende-se também analisar se a proporção dos dados (pares de registros duplicados e não duplicados) na base de treinamento tem influência nos resultados de eficácia.

Em outro experimento a ser realizado, no qual assume-se a existência de um conjunto de dados, seu gabarito e uma base de treinamento rotulada, pretende-se aplicar *Weak Supervision* sobre o conjunto de dados e adicionar estes novos rótulos à base de treinamento, verificando se com o aumento do tamanho desta base de treinamento, houve melhora na eficácia da tarefa de RE. Finalmente, pretende-se aplicar a solução proposta em bases de dados de domínio incomum, para as quais não existem bases de treinamento. As bases de treinamento geradas poderão ser utilizadas por pesquisadores cuja a área seja igual a área destas bases de dados.

Finalmente, existe um planejamento para submissão de um artigo no *Journal of Computer Science and Technology* (JCST) na seção especial *Learning from Small Samples* cujo prazo da chamada finaliza em 20 de Outubro de 2020.

Referências

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

Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., and Stefanidis, K. (2019). End-to-end entity resolution for big data: A survey. *arXiv*.

Dong, X. L. and Rekatsinas, T. (2018). Data integration and machine learning: A natural synergy. In *SIGMOD '18 Proceedings of the 2018 International Conference on Management of Data*, pages 1645–1650.

dos Santos, J. B., Heuser, C. A., Moreira, V. P., and Wives, L. K. (2011). Automatic threshold estimation for data matching applications. *Information Sciences*, pages 2685–2699.

Jureka, A., Hongb, J., Chia, Y., and Liuc, W. (2017). A novel ensemble learning approach to unsupervised record linkage. *Information Systems 71*, pages 40–54.

Köpcke, H., Thor, A., and Rahm, E. (2010). Evaluation of entity resolution approaches on real-world match problems. In *Proceedings of the VLDB Endowment 3(1)*, pages 484–493.

Primpeli, A., Bizer, C., and Keuper, M. (2020). Unsupervised bootstrapping of active learning for entity resolution. *arXiv*.

Ratner, A., Bach, S. H., and Ehrenberg, H. (2017). Snorkel: Rapid Training Data Creation with Weak Supervision). *Proceedings VLDB Endowment. 11(3):*, page 269–282.

Ratner, A., Sa, C. D., Wu, S., Selsam, D., and Re, C. (2016). Data programming: Creating large training sets, quickly. In *Advances in neural information processing systems*, pages 3567–3575.

Whang, S. E. and Garcia-Molina, H. (2010). Entity resolution with evolving rules. In *Proceedings of the VLDB Endowment*.

Zhou, Z.-H. (2017). A brief introduction to weakly supervised learning. *National Science Review 5 (1)*, pages 44–53.

DMless: Uma Abordagem para Gerenciamento de Dados em Ambientes Serverless

João Paulo Vital Santos[1]
Orientador: Flávio R. C. Sousa[1]

[1] Mestrado e Doutorado em Ciência da Computação (MDCC)
Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brasil

paulovital@alu.ufc.br, flaviosousa@ufc.br

Nível: Mestrado
Ingresso: Fev/2019
Exame de Qualificação: Maio/2020
Previsão para defesa: Mar/2021
Etapas concluídas: revisão da literatura, definição do problema e solução
Próxima etapas: Conclusão da implementação e avaliação

Resumo. *A computação serverless é uma tendência de tecnologia destinada a fornecer elasticidade transparente e preços em milissegundos. No entanto, esse modelo de computação requer grandes mudanças tecnológicas, especialmente em SGBDs, pois no ambiente serverless todas as funções têm duração máxima, quantidade fixa de memória e ausência de armazenamento local persistente. Além disso, existe um trade-off entre a latência e o custo que deve ser considerado no uso das plataformas serverless. Para superar estas limitações, este trabalho propõe DMless, uma abordagem para o gerenciamento de dados construída inteiramente sobre funções serverless. Esta abordagem utiliza uma estratégia de endereçamento de funções e combina diferentes tipos de armazenamento para melhorar a latência e reduzir os custos.*

Palavras-chaves*: Serverless, Armazenamento Efêmero, Modelo de Ator, Endereçamento de Função.*

1. Introducão

A computação *serverless* é uma tendência recente da tecnologia destinada a fornecer elasticidade transparente e preços em milissegundos [Jonas et al. 2019]. Seu princípio de funcionamento consiste em escalonar recursos de computação para executar funções *stateless* fornecidas por programadores e acionadas mediante eventos. As plataformas de *Function as a Service* (FaaS) oferecem poder computacional capaz de escalonar para centenas ou até milhares de funções em segundos ou minutos, uma flexibilidade difícil de alcançar até mesmo por modernos sistemas de gerenciamento de banco de dados (SGBDs) [Schleier-Smith 2019]. Tal elasticidade resulta da estrita separação entre computação e armazenamento, um princípio de design arquitetural cada vez mais popular na nuvem [Sreekanti et al. 2020]. Decorre, porém, que as instâncias de execução das funções sejam *stateless*, isto é, sem persistência de dados, exigindo o uso de serviços de armazenamento externo para a troca de estado em aplicações *stateful* [Klimovic et al. 2018].

As *cloud functions* são efêmeras, com duração fixa de poucos minutos e não podem ser endereçadas. A exigência de rapidez na execução e a localização arbitrária das funções dificultam a concepção de um mecanismo de endereçamento de baixo *overhead* [Shafiei et al. 2019]. Como consequência, as aplicações se tornam mais sensíveis à latência e mais dependentes de serviços externos para sincronização. O uso da rede é ainda mais necessário quando se observa que as plataformas *serverless* seguem uma arquitetura *data-shipping*, na qual os dados têm de ser transportados até as funções para que ocorra o processamento, aumentando a latência, o consumo de banda e os custos. Funções de nuvem, no entanto, contam com banda de rede bem mais limitada se comparado com máquinas virtuais tradicionais [Hellerstein et al. 2018].

Embora existam opções para a persistência de dados, nenhuma delas apresenta todos os requisitos ideais para uso com *serverless*: i) capacidade de armazenamento efêmero com custo e desempenho satisfatórios; ii) transparentemente provisionado e acessível às *cloud functions* [Jonas et al. 2019]. As principais soluções de armazenamento oferecidas pelas plataformas *serverless* são inadequadas para operações de granularidade fina [Hellerstein et al. 2018, Jonas et al. 2019]. Aplicações interativas e que fazem uso intensivo de dados são prejudicadas devido ao alto volume de operações de I/O [Klimovic et al. 2018]. Como consequência, o uso da tecnologia *serverless* tem se limitado a contextos simples que contemplam tarefas independentes com pouca ou nenhuma interação [Hellerstein et al. 2018].

Trabalhos recentes se empenharam direta ou indiretamente em oferecer soluções para o gerenciamento de dados em contextos *serverless* [Klimovic et al. 2018, Sreekanti et al. 2020, Barcelona-Pons et al. 2019, Shillaker and Pietzuch 2020, Zhang et al. 2019]. As propostas vão desde extensões de linguagens de programação até novas plataformas de execução de *cloud functions*. No entanto, embora existam trabalhos que tenham explorado os recursos das funções como base para suas soluções, somente [Wang et al. 2020] investiu nelas como estratégia de armazenamento, mas apenas como cache de objetos. Em parte, isso ocorre pelas restrições das funções de nuvem das principais plataformas de mercado, a impossibilidade de se fazer alterações na infraestrutura dos provedores e principalmente devido a falta de um mecanismo de endereçamento [Schleier-Smith 2019].

Para superar estas limitações, este trabalho propõe *DMless*, uma abordagem para o gerenciamento de dados construída inteiramente sobre funções *serverless*. Esta abordagem utiliza uma estratégia de endereçamento de funções e combina diferentes tipos de armazenamento para melhorar a latência e reduzir os custos. Assim, os principais objetivos deste trabalho são:

- A abordagem *DMless* para o gerenciamento de dados construída inteiramente sobre funções serverless. Esta abordagem utiliza uma estratégia de endereçamento de funções e combina diferentes tipos de armazenamento para melhorar a latência e reduzir os custos.
- Uma implementação da abordagem proposta e sua arquitetura.
- Uma avaliação experimental da abordagem proposta.

2. Fundamentação Teórica

Funções de nuvem não possuem persistência de dados local entre as execuções [Sreekanti et al. 2020]. Estas funções têm capacidade limitada de memória RAM e contam com espaço em disco para tarefas temporárias[1], mas esses recursos podem ser reclamados a qualquer momento pelo provedor. Também são efêmeras porque executam em contêineres efêmeros relacionados apenas com a sessão em execução e que são destruídos tão logo se tornam desnecessários. Por fim, são de curta duração porque têm limite de tempo de execução (por exemplo, 15 minutos no Amazon Lambda) ao fim do qual a execução é encerrada, inviabilizando o uso de processos de longa duração.

Um dos grandes desafios atuais da computação *serverless* é o gerenciamento de dados. Sem uma abordagem de armazenamento eficiente e com desempenho satisfatório, é quase inviável construir aplicações de propósito geral que dependam de estado. A natureza e as limitações das funções de nuvem, a arquitetura das plataformas *data-shipping*, gargalos de I/O e a inexistência de serviços de armazenamento adequados restringem o alcance da computação *serverless* a um número limitado de casos de uso.

Em contextos orientados a processamento, o modelo *data-shipping* é eficiente. Contudo, não se pode dizer o mesmo acerca de aplicações orientadas a dados, cujo desempenho acaba sendo prejudicado. Adicione-se o fato de que os provedores de FaaS trabalham para maximizar o número de funções por máquina virtual [Wang et al. 2018], o que contribui para aumentar a contenção de recursos, já que todas as funções dividem a largura de banda disponível para a máquina [Sreekanti et al. 2020]. Assim, em aplicações *stateful*, a necessidade do tráfego de dados é maior por conta do *data-shipping*, mas a largura de banda disponível é bem menor.

Vale destacar que mesmo SGBDs elásticos podem se tornar rapidamente gargalos em face da velocidade superior de escalonamento das *cloud functions* visto que estas escalonam para centenas de funções em segundos [Schleier-Smith 2019]. Um serviço de armazenamento ideal para a computação *serverless* deve escalonar ao nível das funções, responder com baixa latência, apresentar um custo viável e ser tarifado conforme o uso [Schleier-Smith 2019]. As opções atuais apresentam latência e custo médios que dificultam seu uso em cenários de intensa manipulação de dados. Ao mesmo tempo, porém, podem ser opções altamente recomendadas como armazenamento permanente, dadas as garantias de disponibilidade e tolerância a falhas que oferecem [Klimovic et al. 2018].

3. DMless

Uma vez que auto escalonamento e pagamento conforme o uso são caraterísticas próprias da computação *serverless*, é interessante investigar se é possível construir um serviço de armazenamento ou uma abordagem que combine serviços atuais tendo como base as próprias funções

[1]No Amazon Lambda a capacidade de memória RAM das funções é definida pelo usuário e varia de 128MB a 3GB em incrementos de 64MB. O processador é selecionado proporcionalmente a isso, alcançando até 1.7 núcleos, e o espaço em disco para tarefas temporárias é de 500MB.

de nuvem. Sendo assim, este trabalho propõe *DMless*, uma abordagem que utiliza funções de nuvem para oferecer armazenamento efêmero, de baixa latência, transparentemente provisionado e com pagamento baseado no uso. Efêmero e de baixa latência porque oferece armazenamento do tipo chave-valor em memória principal, sendo que o armazenamento permanente fica com os serviços de nuvem que já oferecem garantias adequadas. Transparentemente provisionado pois sua capacidade escala para mais ou para menos conforme a plataforma FaaS utilizada e é tarifado de acordo com o uso.

Compreende-se que seja possível explorar funções de nuvem para prover armazenamento ideal para a computação *serverless*, desde que sejam oferecidos: i) um mecanismo de endereçamento de funções de baixo *overhead*; ii) uma estratégia para maximizar a transferência de dados reduzindo o gargalo de I/O e iii) uma abordagem para gerenciar e orquestrar o estado efêmero das funções oferecendo garantias de consistência e tolerância a falhas. Para tanto, *DMless* explora: i) um mecanismo de endereçamento baseado em filas de mensagens; ii) uma técnica para maximizar a vazão que se baseia na alocação de instâncias por máquina virtual e iii) uma estratégia para orquestrar as funções e oferecer garantias de consistência e tolerância a falhas baseada no Modelo de Ator [Barcelona-Pons et al. 2018].

3.1. Funções Endereçadas e com Maior Largura de Banda

Como as funções não são identificáveis, *DMless* utiliza filas de mensagens para tratar o endereçamento delas [Barcelona-Pons et al. 2018]. As mensagens são os eventos que disparam a execução da instância, necessitando apenas garantir que a mesma instância atenderá cada requisição. Sendo assim, *DMless* reduz a concorrência das funções, obrigando o provedor a utilizar a mesma instância para toda execução de uma mesma função.

Para aumentar a transferência de dados e reduzir os gargalos de rede, *DMless* utiliza uma abordagem similar ao trabalho [Wang et al. 2020] como forma de diminuir a retenção de recursos. Quanto maior o número de funções em uma mesma máquina, menor será a largura de rede associada com cada função [Hellerstein et al. 2018].

3.2. Estratégia de Orquestração

Como estratégia para orquestrar as funções e oferecer garantias de consistência, controle de concorrência e tolerância a falhas, *DMless* utiliza o Modelo de Ator [Barcelona-Pons et al. 2018]. O modelo é um paradigma de computação comum em sistemas distribuídos com foco em controle de concorrência e consistência. O estado é encapsulado em cada ator (unidade computacional básica do modelo) com o objetivo de garantir a consistência.

O Modelo de Ator é interessante por facilitar a construção de aplicações distribuídas e por oferecer as garantias de concorrência e consistência exigidas para o armazenamento de dados mesmo que efêmero. Este modelo já foi utilizado em combinação com *serverless* por [Barcelona-Pons et al. 2018] na construção de uma arquitetura utilizando funções, filas e um SGBD para persistir o estado dos atores e tratar tolerância a falhas. Vale ressaltar que não é o foco desta pesquisa identificar qual é a melhor estratégia para orquestrar funções de nuvem.

3.3. Pagamento Conforme o Uso

DMless permite pagamento conforme o uso por uma característica peculiar do FaaS. Uma instância não é automaticamente destruída tão logo termine de atender uma requisição. Sabe-se que os provedores mantém instâncias em execução durante certo tempo para mitigar o impacto

dos *cold starts*[2] [Jonas et al. 2019]. Se durante esse intervalo não houver novas requisições os recursos alocados são reclamados, caso contrário são mantidos. Assim, pode-se recuperar dados armazenados em memória por execuções anteriores; e como os provedores tarifam por execuções, a memória usada para guardar dados não é tarifada até que estes sejam requisitados. Esta possibilidade também foi explorada no serviço de *cache* com pagamento conforme o uso *Infinicache* [Wang et al. 2020].

3.4. Arquitetura

A arquitetura do *DMless* é dividida em três partes principais: Controlador, *Pool* de funções e Camada de Persistência. Através de uma biblioteca de código na aplicação cliente pode-se interagir com o serviço por meio de funcionalidades básicas de persistência de dados, tais como PUT, GET e DELETE. O Controlador concentra as funcionalidades necessárias para gerenciar e orquestrar as funções de nuvem. O *Pool* de Funções é o conjunto de instâncias de função que são exploradas como abordagem de armazenamento e que escalam de forma horizontal pela plataforma FaaS. Por fim, a Camada de Persistência consiste em um serviço de persistência de nuvem, por exemplo um banco de dados chave-valor, para que o estado interno das funções seja persistido. Uma visão geral da arquitetura é apresentada na Figura 1.

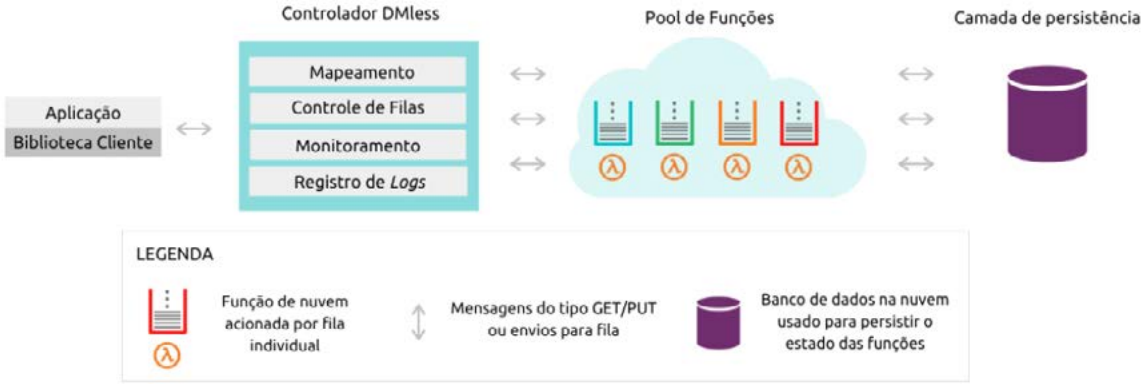


Figura 1. Arquitetura da Abordagem *DMless*

Como exemplo de uso, suponha que uma função *serverless* necessite guardar o retorno do seu processamento para que outras funções (em um encadeamento de funções) possam recuperá-lo. A função cliente usará a operação PUT que será atendida pelo Controlador. Este por sua vez decidirá qual instância do *Pool* de Funções deverá armazenar o valor. É nesta etapa de Mapeamento que se gerencia a relação chave-valor e a localização das instâncias usadas como armazenamento, levando em consideração dados do serviço de Monitoramento. Decidido onde o valor será armazenado, uma solicitação é encaminhada para a fila que representa a função e que, por fim, processará a mensagem guardando o valor em memória. Este dado, por sua vez, será sincronizado periodicamente com um serviço de persistência (como um banco de dados chave-valor, por exemplo). O caminho inverso (operação GET) é mais sugestivo e depende principalmente da gestão de dados mantida pelo Controlador, que deverá saber a localização exata dos dados e encaminhar mensagem de solicitação para a fila correspondente.

É importante destacar que não faz parte do escopo deste trabalho implementar a parte do Controlador com garantias de disponibilidade e tolerância a falhas. O objetivo é explorar

[2] *Cold start* refere-se ao tempo gasto para provisionar uma instância com o necessário antes que se possa executar uma função. A redução desse tempo é um dos grandes desafios do FaaS [Hellerstein et al. 2018].

o potencial das funções de nuvem como estratégia de armazenamento efêmero. Trabalhos futuros poderão investir em uma arquitetura robusta que torne o Controlador mais adequado para ambientes de produção. Por fim, basta considerar que o *Pool* de Funções e a Camada de Persistência *serverless* têm todas estas garantias asseguradas pelo provedor de nuvem.

3.5. Avaliação Experimental

A avaliação consistirá no uso de *microbenchmarks* para comparar o desempenho de I/O do protótipo com sistemas de armazenamento alternativos tais como DynamoDB, S3 e Redis. Um *microbenchmark* para medir a latência de requisição poderá considerar o acesso a 1KB de dados em cada um dos sistemas comparados - conforme usado por [Klimovic et al. 2018]. Outro poderá comparar a taxa de transferência de dados nestes sistemas mediante a requisição de 1MB de dados feita por 100 funções *lambda* concorrentes [Klimovic et al. 2018]. Pretende-se variar a quantidade de memória das funções de nuvem do protótipo, criando diferentes configurações de experimentos, assim como verificar aspectos de custo. Estuda-se também a possibilidade de analisar o desempenho do protótipo em uma aplicação real, a exemplo do que foi feito por [Sreekanti et al. 2020] usando o *Retwis* [Redis 2020], um clone do Twitter útil para avaliar sistemas distribuídos.

4. Trabalhos Relacionados

Esta pesquisa relaciona-se diretamente com trabalhos sobre serviços de armazenamento *serverless* ou plataformas de FaaS que objetivam atender as necessidades de aplicações *stateful* e indiretamente com trabalhos que propõem serviços, técnicas e/ou abstrações com finalidade semelhante.

Em [Klimovic et al. 2018] é proposto *Pocket*, um sistema de armazenamento efêmero voltado para análise de dados que utiliza heurísticas para escolher diferentes *storages*. Já [Sreekanti et al. 2020] propõe *Cloudburst*, uma plataforma de FaaS criada para investigar o ganho de desempenho resultante da proximidade das funções com os dados, usando *cache* local. Os autores [Wang et al. 2020] apresentam *Infinicache*, um sistema de *cache* de objetos construído sobre plataformas *serverless* que explora a memória RAM das funções de nuvem. Por fim, [Zhang et al. 2019] apresentam *Shredder*, um sistema de armazenamento para aplicações *serverless* que explora o uso de funções na camada de armazenamento para reduzir o impacto da arquitetura *data-shipping*.

Estes trabalhos focam na construção de sistemas de armazenamento ou na própria plataforma FaaS para que se possa oferecer suporte adequado a aplicações *stateful*. Diferentemente dos trabalhos relacionados, *DMless* oferece uma abordagem inovadora combinando os recursos já disponíveis nas plataformas FaaS com técnicas de endereçamento de funções associada a diferentes tipos de armazenamento para melhorar o desempenho. Além disso, utiliza o modelo de Ator para tratar aspectos de consistência tornando a solução robusta.

5. Conclusão

Este trabalho apresentou *DMless*, uma abordagem para o gerenciamento de dados em ambientes *serverless* que utiliza funções de nuvem para oferecer armazenamento efêmero, de baixa latência, transparentemente provisionado e pago conforme o uso. Com isso, contribui-se para a ampla utilização de aplicações *stateful* aproveitando as vantagens da tecnologia *serverless*. Como trabalho futuro, pretende-se investigar outras abordagens de endereçamento de funções, bem como propor estratégias para redução de *cold starts*.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

Referências

Barcelona-Pons, D., Ruiz, Á., Arroyo-Pinto, D., and García-López, P. (2018). Studying the feasibility of serverless actors. In *ESSCA'18*, volume 2330 of *CEUR Workshop Proceedings*, pages 25–29.

Barcelona-Pons, D., Sánchez-Artigas, M., París, G., Sutra, P., and García-López, P. (2019). On the faas track: Building stateful distributed applications with serverless architectures. In *Proceedings of the 20th International Middleware Conference*, page 41–54.

Hellerstein, J. M., Faleiro, J. M., Gonzalez, J. E., Schleier-Smith, J., Sreekanti, V., Tumanov, A., and Wu, C. (2018). Serverless computing: One step forward, two steps back. *CoRR*, abs/1812.03651.

Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C., Khandelwal, A., Pu, Q., Shankar, V., Carreira, J., Krauth, K., Yadwadkar, N. J., Gonzalez, J. E., Popa, R. A., Stoica, I., and Patterson, D. A. (2019). Cloud programming simplified: A berkeley view on serverless computing. *CoRR*, abs/1902.03383.

Klimovic, A., Wang, Y., Stuedi, P., Trivedi, A., Pfefferle, J., and Kozyrakis, C. (2018). Pocket: Elastic ephemeral storage for serverless analytics. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 427–444.

Redis (2020). Tutorial: Design and implementation of a simple twitter clone using php and the redis key-value store. `https://redis.io/topics/twitter-clone`. Acesso em: Maio de 2020.

Schleier-Smith, J. (2019). Serverless foundations for elastic database systems. In *CIDR*.

Shafiei, H., Khonsari, A., and Mousavi, P. (2019). Serverless Computing: A Survey of Opportunities, Challenges and Applications. *arXiv e-prints*, page arXiv:1911.01296.

Shillaker, S. and Pietzuch, P. (2020). Faasm: Lightweight isolation for efficient stateful serverless computing.

Sreekanti, V., Wu, C., Lin, X. C., Schleier-Smith, J., Faleiro, J. M., Gonzalez, J. E., Hellerstein, J. M., and Tumanov, A. (2020). Cloudburst: Stateful Functions-as-a-Service. *arXiv e-prints*, page arXiv:2001.04592.

Wang, A., Zhang, J., Ma, X., Anwar, A., Rupprecht, L., Skourtis, D., Tarasov, V., Yan, F., and Cheng, Y. (2020). Infinicache: Exploiting ephemeral serverless functions to build a cost-effective memory cache.

Wang, L., Li, M., Zhang, Y., Ristenpart, T., and Swift, M. (2018). Peeking behind the curtains of serverless platforms. In *Proceedings of the 2018 USENIX Conference on Usenix Annual Technical Conference*, USENIX ATC '18, page 133–145.

Zhang, T., Xie, D., Li, F., and Stutsman, R. (2019). Narrowing the gap between serverless and its state with storage functions. In *Proceedings of the ACM Symposium on Cloud Computing*, SoCC '19, page 1–12, New York, NY, USA. Association for Computing Machinery.

Processamento de consultas analíticas espaciais sobre dados de cidades inteligentes

João Paulo Clarindo dos Santos[1], **Cristina Dutra de Aguiar Ciferri**[1]

[1]Programa de Pós-Graduação em Ciências de Computação
e Matemática Computacional
Instituto de Ciências Matemáticas e Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos, SP – Brasil

jpcsantos@usp.br, cdac@icmc.usp.br

Nível: Mestrado
Ingresso no programa: Março/2019
Etapas concluídas: créditos em disciplinas; exame de qualificação (apresentação em Setembro/2020); proposta de solução
Época esperada de conclusão e defesa: Março/2021

Abstract. *Spatial data generated by the Internet of Things (IoT) devices are important to assist the decision-making in issues related to smart cities. Spatial Data Warehouses (SDW) can be used to process analytical queries extended with spatial predicates (i.e. SOLAP queries – spatial on-line analytical processing). A smart city produces a huge volume of spatial data, thus processing SOLAP queries over spatial data generated by IoT devices is much expensive. Therefore, the processing of SOLAP queries may benefit from the use of frameworks aimed to provide parallelism and data distribution. Also, spatial analytics systems are developed on top of these frameworks to provide support to spatial data. In this paper, we fill a gap in the literature by investigating the processing of SOLAP queries to support the decision-making in the context of IoT and smart cities, using parallel and distributed processing and spatial analytics systems. We aim to introduce the following contributions: (i) specification of an architecture for a SDW environment in the context of IoT and smart cities; (ii) development of new methods related to the SOLAP query processing, using as basis the proposed architecture; and (iii) validation of the architecure and methods using real spatial data obtained from IoT devices.*

Palavras-Chave. *IoT, SOLAP,* data warehouses *espaciais, sistemas analíticos espaciais, cidades inteligentes.*

Trabalho sendo desenvolvido com recursos financeiros do CNPq e FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), processo 2018/22277-8.

1. Introdução

Prover a infraestrutura necessária para comportar uma grande quantidade de pessoas em cidades pode ser um desafio para o poder público e empresas, já que necessitam de mecanismos de monitoramento de recursos para auxiliar a tomada de decisão. Logo, surgiu o conceito de cidades inteligentes, que são cidades cujas infraestruturas utilizam tecnologias visando o crescimento urbano e social. Exemplos são os dispositivos de Internet das Coisas (ou *Internet of Things* (IoT)), que são dispositivos interconectados a fim de fornecer uma base de informações [Yeh 2017, Atzori et al. 2017].

A tecnologia de IoT em uma cidade inteligente pode ser aplicada, por exemplo, no transporte público, cuja frota contém sensores que coletam dados relativos à quantidade de passageiros, tipo de veículo (ônibus, vans, etc.), rota realizada e velocidade máxima, visando aprimorar as linhas existentes. Outro exemplo de aplicação consiste em uma rede de sensores de poluentes espalhados pela cidade que coletam dados relativos à qualidade do ar naquela região, estimulando a melhora da qualidade do ar. Outras aplicações de dispositivos IoT em uma cidade inteligente incluem controle de tráfego, análise do consumo de água e controle de energia de fontes renováveis. [Atzori et al. 2017, Eldrandaly et al. 2019].

Consultas sobre dados gerados por dispositivos IoT em uma cidade inteligente podem auxiliar na tomada de decisão de gestores. Por exemplo, no cenário de dispositivos instalados no transporte público, pode se determinar "*quantos passageiros foram transportados no último mês, por tipo de veículo, por rota, por região*", sendo os resultados obtidos exibidos em um mapa. Uma consulta como esta pode ser respondida a partir da extração, transformação e carga dos dados em um *data warehouse* (DW), que é uma base de dados histórica, orientada a assunto, integrada e não volátil [Kimball et al. 2011]. Os dados são organizados como assuntos de interesse que podem ser analisados a partir de um conjunto de dimensões e que oferecem subsídios para o processamento analítico de consultas (*on-line analytical processing* – OLAP) [Chaudhuri and Dayal 1997].

Dispositivos IoT, além de gerar dados convencionais, tendem a gerar dados espaciais, que são composições que representam a geometria de objetos espaciais. Essas geometrias são usualmente representadas por pontos, linhas e polígonos, ou combinações destes [Güting 1994]. Nesse contexto, as consultas OLAP são limitadas por não considerarem os relacionamentos espaciais. Logo, surgiu o conceito de DW Espacial (DWE), que estende um DW convencional para lidar com dimensões não-espaciais, espaciais ou com ambas. Sobre o DWE incidem consultas Spatial OLAP (SOLAP), as quais oferecem suporte às consultas espaciais [Rivest et al. 2001].

No contexto de cidades inteligentes, DWE é muito volumoso, pois os dispositivos geram muitos dados espaciais [Bonomi et al. 2014]. Isso introduz problemas em relação ao processamento de consultas, uma vez que consultas SOLAP são caras pelo alto custo da junção-estrela e do processamento dos predicados espaciais [Rivest et al. 2001]. Logo, um DWE pode ser favorecido pelo uso de um *framework* de processamento paralelo e distribuído, como Hadoop e Spark, visando diminuir o custo do processamento dos dados espaciais. Este *framework* pode ser inserido em uma nuvem, permitindo a flexibilidade de acordo com a demanda exigida pela cidade inteligente. Entretanto, ele não oferece suporte nativo para o processamento de predicados espaciais, exigindo o uso de sistemas

analíticos espaciais (SAEs), que são expansões que permitem consultas e indexação destes dados espaciais [Castro et al. 2019].

A partir de revisão sistemática (seção 3), não foi detectado um trabalho que relacione o processamento de dados espaciais obtidos de dispositivos IoT no contexto de cidades inteligentes em um DWE inserido em um ambiente paralelo e distribuído. Logo, este projeto de mestrado tem como objetivo investigar essa limitação existente na literatura, com as seguintes contribuições: (i) proposta de uma arquitetura relacionada a temática; (ii) proposta de métodos voltados ao processamento da consulta SOLAP considerando a arquitetura proposta; e (iii) validação da arquitetura e dos métodos propostos sobre dados reais obtidos de dispositivos IoT. O trabalho a ser desenvolvido, embora possua enfoque no contexto de cidades inteligentes, pode ser aplicado a qualquer tipo de dado espacial gerado por uma rede de dispositivos IoT, como exemplo dados espaciais gerados por *smartphones* e que são disponibilizados por mídias sociais.

Este artigo está estruturado da seguinte forma. Na seção 2 é apresentada a fundamentação teórica. Na seção 3 é descrita a revisão sistemática. Na seção 4 é detalhado o estado atual de desenvolvimento do trabalho. Por fim, na seção 5 são descritas as considerações finais e as próximas atividades a serem desenvolvidas.

2. Fundamentação Teórica

2.1. Rede de dispositivos IoT e Computação em Névoa

Existem vários dispositivos IoT como sensores de umidade e pressão, sensores de poluição, *tags* e Arduino. Para criar uma rede que conecte estes dispositivos a um servidor que possa manipular os dados gerados, utilizam-se várias tecnologias de transmissão, como *Radio Frequency IDentification* (RFID), *Global Positioning System* (GPS) e Wi-Fi [Atzori et al. 2017]. O envio de dados diretamente para um servidor próprio ou para um ambiente de computação em nuvem pode se mostrar ineficiente, devido à alta latência na transferência entre os dispositivos [Eldrandaly et al. 2019].

Com isso, [Bonomi et al. 2014] propuseram o paradigma computação em névoa (*fog computing*), que provê serviços de processamento, armazenamento e distribuição de dados próximos à borda da rede. Estes serviços são executados na camada da névoa, que consiste de dispositivos como roteadores, *gateways* e servidores locais capazes de processar, transmitir e armazenar temporariamente os dados recebidos pelos dispositivos IoT, possibilitando operações de extração, transformação e carga (ETL) e processamento analítico em tempo real. As consultas que demandam maior poder de processamento são realizadas na camada da nuvem, assim como é nesta camada que os dados históricos são armazenados. Algumas vantagens do uso de computação em névoa incluem a ampla distribuição geográfica dos serviços, redes de dispositivos distribuídos em larga escala, interações em tempo real, predomínio de acesso sem fio e heterogeneidade da rede de dispositivos IoT.

2.2. Dados espaciais e relacionamentos espaciais

Dados espaciais são componentes que representam a geometria de objetos espaciais. Estes dados podem ser categorizados na forma vetorial ou *raster*. Os dados espaciais podem se relacionar de forma métrica, topológica ou direcional [Güting 1994]. Em um DWE, além de operações OLAP convencionais, também são processadas consultas es-

paciais [Rivest et al. 2001]. As consultas espaciais são consultas nas quais ao menos um de seus predicados envolve um relacionamento espacial. Essas consultas podem se mostrar complexas, sendo necessário o uso de índices espaciais para execução eficiente dessas consultas. Uma estrutura popular de indexação é a R-*tree*, derivada da B^+-*tree* [Gaede and Günther 1998].

3. Revisão Sistemática

A revisão sistemática refere-se a uma metodologia de pesquisa que visa reunir e avaliar as evidências disponíveis referentes a um determinado tópico a partir de passos, que vão desde a formulação de questões de pesquisa até a síntese dos resultados encontrados [Biolchini et al. 2005]. A revisão realizada propôs identificar publicações que atendessem às seguintes questões de pesquisa: (i) Como grandes quantidades de dados espaciais são gerados e manipulados por dispositivos IoT?; (ii) Como os dados espaciais gerados por dispositivos IoT podem ser gerenciados em um ambiente SOLAP?; (iii) Como consultas SOLAP podem auxiliar na tomada de decisão no contexto de cidades inteligentes?; (iv) Como um DW pode ser utilizado em ambientes de computação em névoa?; e (v) Existem estudos que relacionam SOLAP e computação em névoa no contexto de cidades inteligentes a fim de auxílio na tomada de decisão? Foram consideradas as fontes de busca IEEEXplore DL, ACM DL e Elsevier ScienceDirect.

As *strings* de busca utilizadas foram as seguintes (a) *"spatial data" AND ("internet of things" OR iot) AND "big data")*; (b) *(solap OR sdw OR "spatial data warehouse") AND ("internet of things" OR "iot" OR "smart cities")*; e (c) *("data warehouse" OR iot) AND "fog computing"*. Foram retornados 100 trabalhos, sendo selecionados cinco trabalhos que satisfaziam às questões de pesquisa e aos critérios de seleção, os quais incluíram trabalhos em inglês ou português e cuja publicação foi feita entre os anos de 2016 e 2020. Adicionalmente, dois trabalhos foram incluídos para a sintetização de resultados a partir de indicação de especialista. Esses dois trabalhos, embora não atendam ao critério de pesquisa relacionado ao período de tempo, possuem potencial para responder às questões de pesquisa (ii) e (iv). Os trabalhos foram agrupados em três grupos de estudo associados às *strings* de busca (a), (b) e (c), conforme ilustrado na Tabela 1.

Tabela 1. Tópicos abordados nos trabalhos selecionados na revisão sistemática

Trabalho	Grupo de Estudo	Dados Espaciais	SOLAP / OLAP	IoT	Cidades Inteligentes	Computação em Névoa	Satisfaz à Questão de Pesquisa
[Eldrandaly et al. 2019]	(a)	✓	✗	✓	✓	✓	(i)
[Iyer and Stoica 2017]	(a)	✓	✗	✓	✗	✗	(i)
[Jo et al. 2019]	(a)	✓	✗	✓	✗	✗	(i)
[Liu et al. 2019]	(b)	✓	SOLAP	✗	✓	✗	(iii)
[Khakimov et al. 2018]	(b)	✗	OLAP	✓	✗	✓	(iv)
[Yuan and Zhao 2012]	(c)	✓	SOLAP	✓	✗	✗	(ii)
[Bonomi et al. 2014]	(c)	✗	OLAP	✓	✓	✓	(iv)
Abordagem proposta	-	✓	SOLAP	✓	✓	✓	(v)

Embora os trabalhos analisados satisfaçam ao menos uma questão de pesquisa, nenhum deles responde à questão de pesquisa (v), que relaciona SOLAP e IoT no contexto de cidades inteligentes. Nos trabalhos de [Bonomi et al. 2014, Eldrandaly et al. 2019] são propostas arquiteturas que relacionam dados gerados por dispositivos IoT e computação em névoa. Entretanto, essas arquiteturas carecem de processamento analítico em um DWE inserido em um ambiente de processamento paralelo e distribuído. Adicionalmente, o trabalho de [Yuan and Zhao 2012] possui como objetivo a criação de um ambiente SOLAP no contexto de IoT, sendo o que possui maior aproximação com objetivo do mes-

trado. Entretanto, este trabalho não inclui os conceitos mais recentes em relação à IoT, como computação em névoa e processamento paralelo e distribuído. O trabalho a ser desenvolvido no mestrado pretende cobrir esta lacuna existente na literatura, conforme listado na Tabela 1.

4. Proposta e Estágio Atual de Desenvolvimento

4.1. Descrição da arquitetura

Para suprir a lacuna existente nas literatura e dado o objetivo deste trabalho, foi proposta uma arquitetura que relaciona dados obtidos por dispositivos IoT no contexto de cidades inteligentes com um DWE inserido em um ambiente de processamento e armazenamento paralelo e distribuído. Esta arquitetura, ilustrada na Figura 1, possui três camadas: terminal, processamento na névoa e processamento na nuvem.

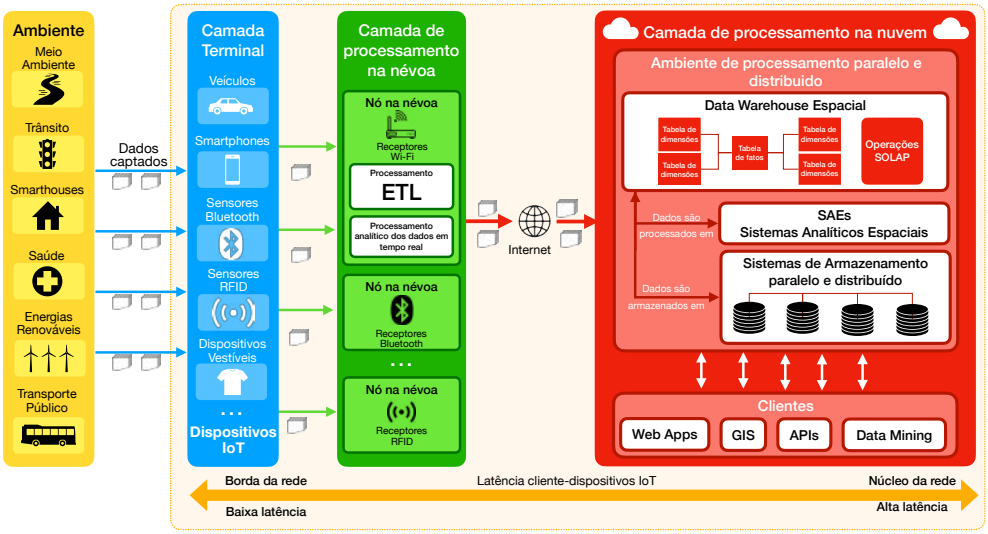


Figura 1. Visão geral da arquitetura proposta

Na camada terminal, os dados são coletados do ambiente a partir de dispositivos IoT, como sensores, *smartphones*, veículos e dispositivos vestíveis, dentre outros. Esses dispositivos conectam-se com a camada de processamento da névoa, que contém receptores com diversos protocolos de comunicação como exemplo Wi-Fi, RFID, Bluetooth e que possuem poder limitado de processamento e armazenamento dos dados. Estes receptores, que são os nós da névoa, são responsáveis pelas operações de ETL e pelo processamento analítico dos dados em tempo real, pois estes dispositivos encontram-se próximos à borda da rede, ocasionando na baixa latência entre o nó na névoa e o usuário.

Com relação ao processamento analítico dos dados históricos, os dados, após passarem pela etapa de ETL, são enviados para a camada de processamento em nuvem. Nesta camada, os dados são armazenados em um sistema de armazenamento paralelo e distribuído, dispostos em um DWE. As consultas SOLAP são processadas em SAEs vinculados ao sistema de processamento e armazenamento paralelo e distribuído. Pela natureza escalável da computação em nuvem, a quantidade de nós em um ambiente paralelo e distribuído pode ser aumentado ou diminuído dependendo das demandas de consultas da rede. Por fim, os clientes, como Web Apps e Geographic Information Systems (GIS) realizam consultas no DWE.

4.2. Detalhamento das Atividades

No momento atual do trabalho, está sendo investigada a instanciação da arquitetura com tecnologias disponíveis e de acordo com os objetivos do trabalho. O foco principal da pesquisa é a camada de processamento na nuvem. Portanto, a camada de processamento em névoa prevê o uso de tecnologias já existentes para oferecer subsídios para o trabalho. Essa camada utiliza o Apache Airflow para realizar operações ETL e o PostgreSQL para armazenamento dos dados antes de serem enviados para a nuvem. Na camada de processamento na nuvem, os dados do DWE são manipulados utilizando o *framework* de processamento paralelo e distribuído Spark e armazenados no *Hadoop File System* (HDFS). A escolha das tecnologias descritas foi feita com base nos avanços obtidos pelo grupo de pesquisa no qual o projeto de mestrado se enquadra.

Com relação ao processamento de consultas SOLAP considerando a arquitetura proposta, pretende-se desenvolver métodos para processar essas consultas eficientemente. Os métodos devem considerar os dados espaciais vetoriais e os relacionamentos topológicos de *intersection, enclosure* e *containment*. Para o processamento da junção-estrela, pretende-se usar os avanços descritos no trabalho de [Sangat et al. 2020], que representa o estado-da-arte neste assunto. Para o processamento do relacionamento topológico, pretende-se usar as funcionalidades providas pelo GeoSpark [Yu et al. 2015]. Também pretende-se investigar o projeto de diferentes esquema-estrela para a organização do DWE. Será analisado como estas disposições impactam na execução das consultas espaciais utilizando como base os estudos realizados por [Mateus et al. 2016]. É importante observar que os métodos a serem desenvolvidos visam integrar os componentes da arquitetura.

Para a validação da arquitetura e dos métodos propostos, serão usados *datasets* contendo dados gerados em dispositivos IoT no contexto de cidades inteligentes, fornecidos por portais de dados abertos governamentais e dados das cidades de Aarsus e Brașov gerados por [Ali et al. 2015]. Alguns tipos de dados encontrados nestes *datasets* incluem dados de poluentes, tráfego urbano, energia elétrica e estacionamentos. Os parâmetros para teste de desempenho da arquitetura incluem a definição do volume, ambiente, complexidade e seletividade das consultas, custo de processamento e tempo de execução destas consultas. Os resultados serão publicados em conferências e periódicos relacionados às áreas de banco de dados, geoinformática e IoT.

5. Conclusão

O projeto de mestrado tem como objetivo investigar o processamento de dados espaciais obtidos de dispositivos IoT no contexto de cidades inteligentes em um DWE inserido em um ambiente paralelo e distribuído, com ênfase em SOLAP. Neste contexto, este artigo descreve os conceitos de computação em névoa, dados espaciais e consultas espaciais, além de apresentar uma revisão sistemática relacionada aos temas do trabalho. O artigo também detalha os objetivos da pesquisa e descreve a arquitetura proposta, a qual será usada como base para o desenvolvimento do trabalho. A próxima etapa a ser realizada refere-se ao desenvolvimento dos métodos para o processamento de consultas SOLAP, conforme detalhado na seção 4.2. Na sequência, ocorrerá a validação desses métodos.

Referências

Ali, M. I., Gao, F., and Mileo, A. (2015). CityBench: A configurable benchmark to evaluate RSP engines using smart city datasets. In *LNCS*, volume 9367, pages 374–389. Springer Verlag.

Atzori, L., Iera, A., and Morabito, G. (2017). Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122–140.

Biolchini, J., Gomes Mian, P., Candida Cruz Natali, A., and Horta Travassos, G. (2005). Systematic Review in Software Engineering. Technical report, COPPE/UFRJ.

Bonomi, F., Milito, R., Natarajan, P., and Zhu, J. (2014). Fog computing: A platform for internet of things and analytics. *Studies in Computational Intelligence*, 546:169–186.

Castro, J. P. C., Carniel, A. C., and Ciferri, C. D. A. (2019). A User-centric View of Distributed Spatial Data Management Systems. In *GEOINFO*, pages 80–91.

Chaudhuri, S. and Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1):65–74.

Eldrandaly, K. A., Abdel-Basset, M., and Shawky, L. A. (2019). Internet of Spatial Things: A New Reference Model With Insight Analysis. *IEEE Access*, 7:19653–19669.

Gaede, V. and Günther, O. (1998). Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231.

Güting, R. H. (1994). An introduction to spatial database systems. *The VLDB Journal*, 3(4):357–399.

Iyer, A. P. and Stoica, I. (2017). A scalable distributed spatial index for the internet-of-things. In *ACM SoCC*, pages 548–560.

Jo, J., Joo, I. H., and Lee, K. W. (2019). Constructing national geospatial big data platform: Current status and future direction. In *IEEE WF-IoT*, pages 979–982.

Khakimov, A., Muthanna, A., and Muthanna, M. S. A. (2018). Study of fog computing structure. In *IEEE ElConRus*, pages 51–54.

Kimball, R., Ross, M., Thornthwaite, W., Mundy, J., and Becker, B. (2011). *The Data Warehouse Lifecycle Toolkit*, volume 3. John Wiley & Sons Inc, Hoboken, NJ.

Liu, C., Wu, C., Shao, H., and Yuan, X. (2019). SmartCube: An Adaptive Data Management Architecture for the Real-Time Visualization of Spatiotemporal Datasets. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1.

Mateus, R. C., Siqueira, T. L. L., Times, V. C., Ciferri, R. R., and Ciferri, C. D. A. (2016). Spatial data warehouses and spatial OLAP come towards the cloud: design and performance. *Distributed and Parallel Databases*, 34(3):425–461.

Rivest, S., Bédard, Y., and Marchand, P. (2001). Toward better support for spatial decision making: defining the characteristics of Spatial On-Line Analytical Processing (SOLAP). *Geomatica*, 55(4):539–555.

Sangat, P., Taniar, D., and Messom, C. (2020). Distributed ATrie Group Join: Towards Zero Network Cost. *IEEE Access*, 8:111598–111613.

Yeh, H. (2017). The effects of successful ICT-based smart city services: From citizens' perspectives. *Government Information Quarterly*, 34(3):556–565.

Yu, J., Wu, J., and Sarwat, M. (2015). GeoSpark: A cluster computing framework for processing large-scale spatial data. In *ACM GIS*, pages 1–4, New York.

Yuan, L. and Zhao, J. (2012). Construction of the system framework of Spatial Data Warehouse in Internet of Things environments. In *5th IEEE ICACI*, pages 54–58.

DSAdvisor: Uma Ferramenta para Guiar a Execução de Tarefas Preditivas em Ciência de Dados

José Augusto Câmara Filho[1], **José Maria Monteiro**[1]

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

augustocam95@gmail.com, monteiro@dc.ufc.br

Nível: Mestrado.
Ingresso: Março de 2019.
Previsão de Término: Fevereiro de 2021.
Programa: Programa de Mestrado e Doutorado em Ciência da Computação.
Etapas já concluídas: Pré-Proposta defendida, disciplinas concluídas.
Defesa da Proposta: Dezembro de 2020.

Abstract. *Currently, professionals from the most diverse areas of knowledge need to explore their data repositories in order to extract knowledge and create new products or services. Several tools have been proposed in order to assist the tasks involved in the Data Science life cycle. However, such tools require from their users have specific (and deep) knowledge in the areas of Computing and Statistics. Therefore, the use of these tools is practically unfeasible for a non-specialist professional in Data Science. In this work, we propose a tool, called DSAdvisor, to guide these users in the execution of the main stages presents in the Data Science life cycle. More precisely, DSAdvisor guides these professionals in predictive tasks, that is, involving regression and classification. As main characteristics of the DSAdvisor we can mention: the use of a well-defined script (sequence) of actions, constant feedback, among others.*

Resumo. *Atualmente, profissionais das mais diversas áreas necessitam explorar seus repositórios de dados com a finalidade de extrair conhecimento e assim criar novos produtos ou serviços. Diversas ferramentas têm sido propostas com a finalidade de auxiliar as tarefas envolvidas no ciclo de vida da Ciência de Dados. Todavia, tais ferramentas exigem que seus usuários possuam conhecimentos específicos (e profundos) nas áreas de Computação e Estatística. Portanto, a utilização dessas ferramentas torna-se praticamente inviável para um profissional que não seja especialista em Ciência de Dados. Neste trabalho, propomos uma ferramenta, chamada DSAdvisor, para guiar usuários não especialistas na execução das principais etapas presentes no ciclo de vida da Ciência de Dados. Mais precisamente, a DSAdvisor orienta esses profissionais em tarefas preditivas, ou seja, que envolvem regressão e classificação. Como principais características da DSAdvisor podemos citar: a utilização de um roteiro (sequência) bem definida de ações, feedback constante, dentre outras.*

1. Introdução

Devido a grande quantidade de dados disponíveis atualmente, diversos profissionais, das mais diversas áreas, necessitam explorar seus repositórios com a finalidade de extrair conhecimento e, assim, criar novos produtos ou serviços. Por exemplo, médicos cardiologistas necessitam explorar grandes repositórios de sinais ECG (eletrocardiograma) com a finalidade de predizer a probabilidade de morte súbita de um determinado paciente. Outro exemplo, auditores das Secretarias da Fazenda desejam explorar suas bases de dados com o objetivo de predizer a probabilidade de um determinado contribuinte ser um sonegador de impostos. Neste contexto, o volume e a variedade de dados ultrapassam em muito a capacidade de análise manual dos seres humanos. Contudo, os computadores se tornaram muito mais poderosos e foram desenvolvidos algoritmos que permitem identificar padrões escondidos nesses conjuntos de dados. A convergência desses fenômenos impulsionou o desenvolvimento e a popularização da Ciência de Dados [Provost and Fawcett 2013].

A Ciência de Dados é uma área multidisciplinar que orienta a extração de informações e conhecimento a partir de grandes volumes de dados [Provost and Fawcett 2013]. Mais precisamente, ela trata da coleta, integração, gerenciamento, exploração dos conjuntos de dados e da utilização do conhecimento adquirido com a finalidade tomar decisões, entender o passado/presente, prever o futuro e criar novos serviços/produtos [Ozdemir 2016]. Assim, a Ciência de Dados busca obter novas ideias ("*insights*") que estejam escondidas nesses grandes repositórios.

A Figura 1, ilustra as etapas do ciclo de vida da Ciência de Dados: *Business understading*, *Data understading*, *Data preparation*, *Modeling*, *Evaluation* e *Deployment*. Assim, para obter conhecimentos que estejam escondidos nos dados devemos ser capazes de: (i) entender os problemas a serem solucionados com a utilização das técnicas de mineração de dados, (ii) compreender os dados e os relacionamentos entre eles, (iii) extrair um subconjunto dos dados que possa ser utilizado para solucionar um determinado problema, (iv) criar modelos de aprendizagem automática que possam solucionar o problema investigado, (v) avaliar o desempenho dos modelos criados e (vi) apresentar como esses modelos podem ser utilizados para auxiliar os processos de tomada de decisão [Chertchom 2018]. Desta forma, o domínio de todo o ciclo de vida da Ciência de Dados torna-se praticamente inviável para usuários não especialistas.

Neste contexto, diversas ferramentas têm sido propostas com a finalidade de auxiliar as tarefas envolvidas no ciclo de vida da Ciência de Dados, como, por exemplo: Knime [Berthold et al. 2009], Orange [Demšar et al. 2013], RapidMiner [Hofmann and Klinkenberg 2016] e Weka[Hall et al. 2009]. Todavia, tais ferramentas exigem que seus usuários possuam conhecimentos específicos nas áreas de Computação e de Estatística. Elas foram construídas para serem utilizadas por Cientistas de Dados. Ademais, possuem uma longa curva de aprendizado. Logo, tais *softwares* não são apropriados para usuários não especialistas (médicos, biólogos, geógrafos, auditores, etc).

Neste trabalho, propomos uma ferramenta *web*, chamada **DSAdvisor**, para guiar usuários não especialistas na descoberta de conhecimento e "insights" a partir de grandes volumes de dados. Mais precisamente, o **DSAdvisor** orienta esses profissionais em tarefas que envolvam regressão ou classificação. A classificação tem por finalidade atribuir um rótulo a um determinado item a partir de um conjunto discreto de possibilidades. Já a regressão tem por objetivo prever um determinado valor numérico [Skiena 2017].

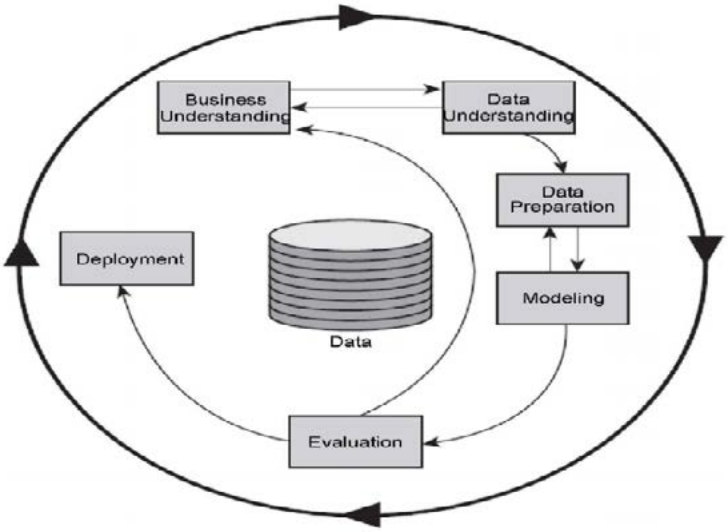


Figura 1. O ciclo de vida da ciência de dados — Fonte: [Chapman et al. 2019]

Como principais características da DSAdvisor podemos citar: o suporte ao entendimento dos dados (utilização de técnicas para descrever e sumarizar o conjunto dos dados, para analisar a distribuição dos dados e para investigar a correlação entre os dados), o suporte à preparação dos dados (divisão do conjunto de dados em treino, teste e validação; utilização de técnicas de detecção de outliers, seleção de atributos e normalização), o *feedback* constante, dentre outras.

O restante deste artigo está organizado da seguinte forma. A seção 2 destaca os trabalhos relacionados. Na seção 3, a ferramenta proposta é apresentada. A metodologia utilizada é discutida na seção 4. Por fim, a seção 5 conclui este trabalho e apresenta direções para pesquisas futuras.

2. Trabalhos Relacionados

Esta seção discute os principais trabalhos relacionados. Para um melhor entendimento, agrupamos esses trabalhos em duas categorias: ferramentas de suporte e *Guidelines*.

2.1. Ferramentas de Suporte

Antes mesmo da popularização da Ciência de Dados, diversas ferramentas foram propostas com a finalidade de auxiliar as tarefas relacionadas a mineração de dados. Tais ferramentas diferem entre si em alguns aspectos, tais como: **usabilidade** da ferramenta, tipo de **licença**,**linguagem** de programação em que foi desenvolvida e suporte ao **entendimento dos dados**. A Tabela 1 ilustra as principais ferramentas encontradas na literatura [Hasim and Haris 2015]. Dentre essas, merecem destaque: KEEL, Knime, Orange, RapidMiner, Tanagra e Weka.

2.2. Guidelines

Guideline é um roteiro que determina o curso de um conjunto de ações que compõem um processo específico, além de um conjunto de boas práticas que devem ser utilizadas durante a realização dessas atividades [Dictionary 2015]. Alguns *guidelines* têm sido propostos com o objetivo de orientar as atividades comumente realizadas em mineração

Lista de *softwares*				
Software	**Usabilidade**	**Licença**	**Linguagem**	**Entendimento dos dados**
KEEL	*Alta*	GPL	Java	Faz
KNIME	*Baixa*	Outra	Java	Faz
RapidMiner	*Alta*	GPL	Java	Parcial
Orange	*Baixa*	GPL	C++, Python	Não faz
TANAGRA	*Alta*	Outra	C++	Faz
Weka	*Baixa*	GPL	Java	Parcial

Tabela 1. Características gerais dos *softwares* de mineração de dados. Fonte: [Hasim and Haris 2015] (adaptada)

de dados. [Melo et al. 2019] propõe um roteiro prático para dar suporte a previsão de mudanças em *softwares* utilizando modelos preditivos. Já em [Luo et al. 2016], os autores buscam obter um conjunto de diretrizes a cerca do uso de modelos preditivos em ambientes clínicos, afim de assegurar que as atividades sejam corretamente executadas e relatadas.

3. A solução proposta

A ferramenta DSAdvisor tem por finalidade guiar usuários não especialistas na execução das principais etapas do ciclo de vida da Ciência de Dados. Mais precisamente, orienta esses profissionais em tarefas preditivas, ou seja, que envolvem regressão e classificação. A DSAdvisor tem como principais características: o suporte ao entendimento dos dados, o suporte à preparação dos dados, o *feedback* constante, dentre outras.

3.1. Visão geral

A DSAdvisor adota um roteiro de ações adaptado do *guideline* proposto por [Melo et al. 2019]. A Figura 2 ilustra as etapas utilizadas pela DSAdvisor. A seguir, descrevemos cada uma dessas fases.

Etapa 1 - Entrada de Dados. Inicialmente, o usuário faz o *upload* dos dados a serem explorados. Para isso, ele deve indicar um arquivo em formato csv. Após o *upload*, a DSAdvisor exibe uma amostra dos dados contendo 20 linhas.

Etapa 2 - Análise Exploratória de Dados. Nesta etapa, a DSAdvisor aplica diferentes técnicas para descrever e sumarizar o conjunto dos dados. Por exemplo, são exibidas diferentes medidas de dispersão, tais como: média, quantidade de instâncias, valor máximo, valor mínimo, percentis, desvio padrão, etc. É exibido também a quantidade e o percentual de valores faltantes. Adicionalmente, a DSAdvisor utiliza diferentes métodos para analisar a distribuição dos dados, tais como: Cramér–Von Mises, D'Agostino's K-squared, Lilliefors, Shapiro-Wilk e Kolmogorov-Smirnov. Em seguida, a DSAdvisor exibe o resultado de diferentes coficientes de correlação (Pearsom, Spearman e V de Cramer). Por fim, o usuário deve informar o tipo de problema a ser analisado, indicar a variável dependente e definir a proporção dos conjuntos de treino, validação e teste.

Etapa 3 - Análise de *Outliers*. Na terceira etapa, a DSAdvisor utiliza o método de distância interquartílica ajustada [Hubert and Vandervieren 2008] para identificar valores discrepantes presentes no conjunto de treino.

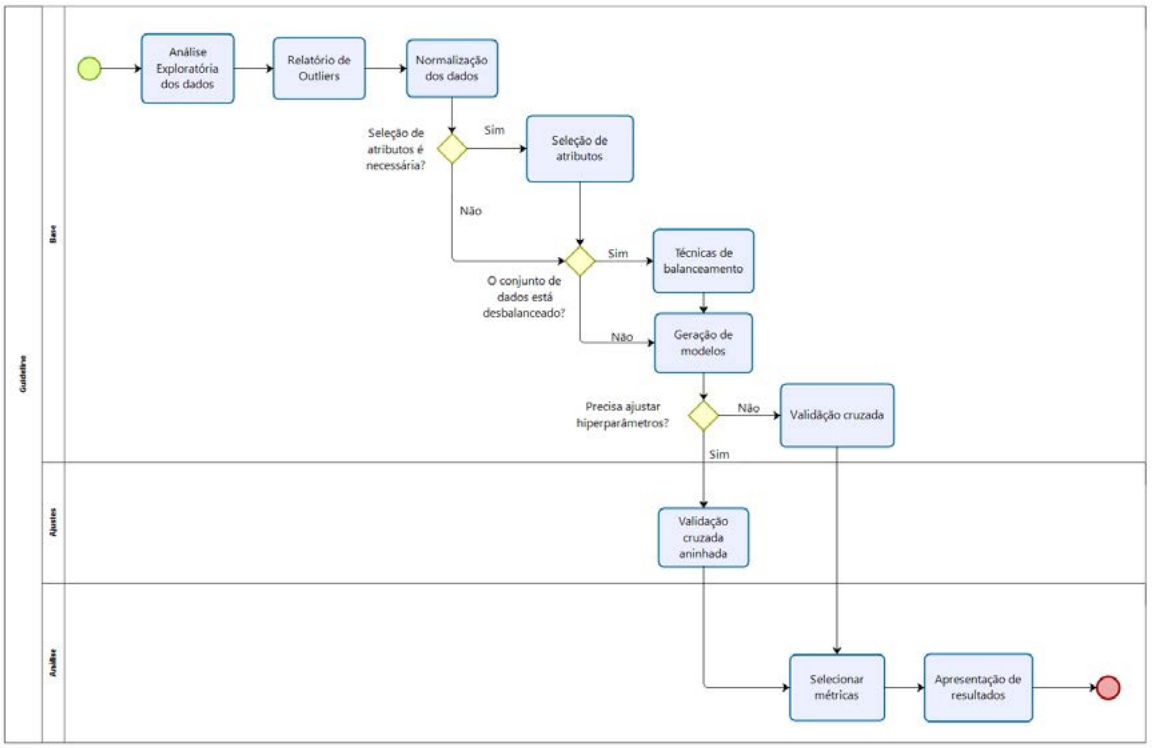


Figura 2. Roteiro de ações utilizado pela DSAdvisor. Fonte: adaptado de Melo et al.(2019)

Etapa 4 - Normalização dos Dados. A ferramenta utiliza a fórmula *z-score* para normalizar as variáveis numéricas do conjunto de treino.

Etapa 5 - Seleção de Atributos. Nesta etapa, a DSAdvisor aplica diferentes métodos de seleção de atributos do tipo *filters*. Em seguida, os resultados de cada método são combinados, seguindo a abordagem proposta em [Melo 2020], com a finalidade de gerar um resultado final, ou seja, o conjunto dos atributos que será sugerido ao usuário para a construção dos modelos preditivos.

Etapa 6 - Balanceamento dos Dados. Aqui, a ferramenta irá analisar e sugerir que técnicas de balanceamento de dados podem ser utilizadas de acordo com o conjunto de dados fornecido pelo usuário. A DSAdvidor explora diferentes técnicas de *Oversampling* e *Undersamplig*.

Etapa 7 - Geração dos Modelos Preditivos. Na Etapa 7, o DSAdvisor solicita ao usuário que especifique o tipo do problema em investigação: regressão ou classificação. Dependendo do tipo do problema, diferentes algoritmos são selecionados e os modelos construídos. Para isso, a ferramenta proposta irá utilizar soluções de "AutoML" já existentes.

Etapa 8 - Validação Cruzada. A validação cruzada é utilizada para atestar o desempenho do modelo obtido durante a fase de treinamento.

Etapa 9 - Validação Cruzada Aninhada. Após definir uma *grid-search* em uma região específica, a validação cruzada aninhada deve ser utilizada para estimar o erro de generalização do modelo subjacente e buscar ajustar os valores dos hiperparâmetros.

Etapa 10 - Seleção de Métricas. Para avaliar o modelo de Aprendizado de Máquina é

necessário selecionar as métricas de desempenho apropriadas, de acordo com o tipo do problema (classificação ou regressão) e o balanceamento dos dados. Considerando esses dois aspectos, a DSAdvisor seleciona automaticamente às métricas mais indicadas.

Etapa 11 - Apresentação dos Resultados. Nesta última etapa, a DSAdvisor apresenta os resultados obtidos, juntamente com as métricas e hiper parâmetros utilizados.

3.2. Implementação

A ferramenta DSAdvisor está sendo implementada utilizando a linguagem **Python** [van Rossum 1995] como *back-end*. O *front-end* está sendo desenvolvido em *HTML 5*, *CSS3*, *Bootstrap*, *JS* e o *Framework Web* **Flask** [Grinberg 2018]. Até o presente momento, as etapas de 1 a 4 já foram implementadas.

4. Metodologia

Este trabalho está dividido em 7 fases:

1. Levantamento bibliográfico e teórico, tendo como objetivo o aprofundamento do conhecimento acerca de técnicas e algoritmos a serem utilizados na ferramenta;
2. Estudo comparativo das ferramentas existentes mais populares;
3. Implementação da ferramenta proposta;
4. Realização de testes de usabilidade com usuários não especialistas;
5. Entrevistas com usuários não especialista;
6. Análise dos resultados obtidos nos testes de usabilidade e entrevistas;
7. Publicação dos resultados obtidos em periódicos ou conferências internacionais.

5. Conclusões e Trabalhos Futuros

Neste trabalho, propomos uma ferramenta, chamada DSAdvisor, para guiar usuários não especialistas na execução das principais atividades presentes no ciclo de vida da Ciência de Dados. Mais precisamente, ela orienta esses profissionais em tarefas preditivas, ou seja, que envolvem regressão e classificação. Diferentemente das ferramentas existentes, a DSAdvisor fornece amplo suporte ao entendimento e preparação dos dados. Além disso, a DSAdvisor segue um *guideline* bem definido e fornece um *feedback* constante. Vale destacar que a ferramenta proposta ainda está em desenvolvimento. Portanto, como atividades futuras destacamos a realização de destes de usabilidade e entrevistas com usuários não especialistas. A princípio, auditores da Secretaria de Fazenda do Estado do Maranhão (SEFAZ-MA).

Referências

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., and Wiswedel, B. (2009). Knime-the konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1):26–31.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., SHEA-RER, C., and Wirth, R. (2019). Crisp-dm 1.0 step-by-step data mining guide/crisp-dm consortium. 2000.

Chertchom, P. (2018). A comparison study between data mining tools over regression methods: Recommendation for smes. In *2018 5th International Conference on Business and Industrial Research (ICBIR)*, pages 46–50. IEEE.

Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., et al. (2013). Orange: data mining toolbox in python. *the Journal of machine Learning research*, 14(1):2349–2353.

Dictionary, C. (2015). Cambridge dictionaries online.

Grinberg, M. (2018). *Flask web development: developing web applications with python*. "O'Reilly Media, Inc.".

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Hasim, N. and Haris, N. A. (2015). A study of open-source data mining tools for forecasting. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, pages 1–4.

Hofmann, M. and Klinkenberg, R. (2016). *RapidMiner: Data mining use cases and business analytics applications*. CRC Press.

Hubert, M. and Vandervieren, E. (2008). An adjusted boxplot for skewed distributions. *Computational statistics & data analysis*, 52(12):5186–5201.

Luo, W., Phung, D., Tran, T., Gupta, S., Rana, S., Karmakar, C., Shilton, A., Yearwood, J., Dimitrova, N., Ho, T. B., et al. (2016). Guidelines for developing and reporting machine learning predictive models in biomedical research: a multidisciplinary view. *Journal of medical Internet research*, 18(12):e323.

Melo, C. S. (2020). Supporting change-prone class prediction.

Melo, C. S., da Cruz, M. M. L., Martins, A. D. F., Matos, T., da Silva Monteiro Filho, J. M., and de Castro Machado, J. (2019). A practical guide to support change-proneness prediction. In *ICEIS (2)*, pages 269–276.

Ozdemir, S. (2016). *Principles of data science*. Packt Publishing Ltd.

Provost, F. and Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. *Big data*, 1(1):51–59.

Skiena, S. S. (2017). *The data science design manual*. Springer.

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

Sessão de Ferramantas

Demonstrations Sessions

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 35th edition, SBBD, due to COVID-19 and coronavirus pandemic, will have all activities happening online only, from September 28th to October 1st, 2020.

The Demonstrations and Applications Session is organized since 2004 within SBBD. The Demonstrations Session has become an important venue for sharing prototype data management systems with the SBBD community. The session aims to reveal 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 include a new category of demo papers: distinguished demos. We invited two well-established demonstrations to be part of the 2020 track. Both invited demos present data modeling tools: brModelo and TerraER. We have also selected four interesting demo papers from submissions. As expected, two of them are about problems related to COVID-19. Each paper was evaluated by three reviewers selected from a committee of 24 researchers from both academia and industry.

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

Finally, we would like to thank the SBBD 2020 organizers for accepting the challenge of organizing an online event and providing the Demonstration and Application Session infrastructure.

We hope you all enjoy SBBD Demonstration and Application online!

Denio Duarte (UFFS)
Program Chair – SBBD 2018 – Demos and Applications

Rastreador de sintomas da COVID19

Ticiana L. Coelho da Silva[1], **Marianna Gonçalves F. Ferreira**[1], **Regis Pires Magalhães**[1], **José Antônio F. de Macêdo**[1], **Natanael da Silva Araújo**[1]

[1]Insight Data Science Lab – Universidade Federal do Ceará (UFC)
Caixa Postal 60.440-900 – Fortaleza – CE – Brasil

{ticianalc, marianna, regis, jose.macedo, natanaelsilva}@insightlab.ufc.br

Abstract. *The pandemic caused by coronavirus has fueled the need for technological solutions capable of capturing and monitoring data in an automatic, agile and secure way. The coronavirus on-call service system made available to the population of the Ceará State has automatic symptom recognition technology through Natural Language Processing (NLP). The tracker proposed in this work, called Sintomatic, is a neural network that processes texts, capturing symptoms in messages exchanged between the citizen of Ceará and the nurse/doctor at the Plantão Coronavirus. In addition, Sintomatic identifies and captures mental health behaviors, such as: anxiety, distress and trends in depression.*

Resumo. *A pandemia causada por coronavírus fomentou a necessidade de soluções tecnológicas capazes de capturar e monitorar dados de forma automática, ágil e segura. A Plataforma de Atendimento Plantão Coronavírus disponibilizada para a população do Estado do Ceará possui tecnologia de reconhecimento automático de sintomas por meio do Processamento de Linguagem Natural (NLP). O rastreador proposto neste trabalho, chamado de Sintomatic, é uma rede neural que processa textos, capturando sintomas em mensagens trocadas entre o cidadão cearense e o enfermeiro/médico no Plantão Coronavírus. Além disso, o Sintomatic identifica e captura comportamentos de saúde mental, como: ansiedade, angústia e tendências de depressão.*

1. Introdução

Diante do cenário causado pela pandemia por coronavírus e o acometimento no Estado do Ceará desde março de 2020, surgiu a demanda do desenvolvimento soluções tecnológicas que fossem capazes de capturar e monitorar dados de forma automática, ágil e segura, pois pouco sabia-se sobre o comportamento a e evolução do vírus.

Gestores de saúde e tomadores de decisão necessitavam de dados para mitigar um período de desconhecimento e incertezas. Conhecer os padrões da doença era crucial para elaborar protocolos de saúde eficientes. Para isso, era preciso reconhecer sintomas e comportamentos de saúde mental da população acometida.

Uma das soluções desenvolvidas e disponibilizadas para a população no Estado do Ceará foi o Plantão Coronavírus, uma plataforma com mecanismos de triagem que, no primeiro momento utiliza um *chatbot* para interagir com o paciente a fim de classificar seu estado de saúde em uma das três categorias: verde, amarelo e vermelho, sendo o nível de criticidade da saúde do paciente leve, moderada ou grave, respectivamente. Após a

triagem com o chat de teleatendimento do Ceará, ele pode ser encaminhado para uma tele consulta com um profissional de saúde, a depender do seu quadro clínico.

As interações entre os pacientes e os profissionais de saúde por meio do Plantão Coronavírus geraram muitos dados que precisavam ser minerados, analisados e transformados em informação de valor.

A Secretaria de Saúde do Estado do Ceará precisava rastrear os sinais da doença e era inviável executar essa tarefa manualmente por meio da leitura de milhares de relatos. Dessa forma, uma solução automatizada e inteligente para classificar os padrões da COVID19 era imprescindível.

Este trabalho mapeou a identificação de sintomas em texto como um problema de reconhecimento de entidade (em inglês, *Named Entity Recognition – NER*). *NER* corresponde à capacidade de identificar as entidades nomeadas nos documentos e rotulá-las em classes definidas de acordo com o tipo de entidade [da Silva et al. 2019]. De forma geral, o robô de captura de sintomas possui uma rede neural que é capaz de reconhecer entidades. Neste trabalho, uma entidade é um sintoma.

O mecanismo de captura de sintomas perpassa por todo o processo de triagem com o *chatbot*, até o tele atendimento com o profissional de saúde. O robô de captura de sintomas, chamado de Sintomatic, é a principal contribuição deste trabalho.

Sintomatic é uma tecnologia que consome os dados da plataforma Plantão Coronavírus, e então é capaz de processar e identificar os sintomas contidos nos textos em linguagem natural, utilizando Processamento de Linguagem Natural (PLN), tecnologia largamente utilizada para ajudar computadores a entender a linguagem do ser humano. O link [1] apresenta uma breve demonstração do Sintomatic.

Este tipo de inteligência foi essencial para identificar padrões de sinais da doença, bem como novos sintomas ou sintomas raros, que ainda não haviam sido mapeados pelos profissionais de saúde, e, dessa forma, acompanhar a evolução dos achados da COVID19 ao longo dos dias.

O processo de reconhecimento de entidades foi realizado completamente automático, sendo destacado como um diferencial frente aos trabalhos relacionados [Tarcar et al. 2020] que apresentou F1 de 78,5% e [Neumann et al. 2019] que apresentou 84,94% de F1 para o modelo de descoberta de sintomas [2], enquanto o Sintomatic tem F1 igual a 85,66%. O cálculo do F1 mede a acurácia do conjunto de teste, combinando as métricas de *recall* e de *precision*.

Nas seções seguintes, serão abordadas a metodologia usada na construção do Sintomatic e os cenários de demonstração. E por fim, a conclusão deste artigo.

2. Sintomatic

O Sintomatic é um modelo computacional, que foi desenvolvido com o objetivo de auxiliar a Secretaria de Saúde do Estado do Ceará no acompanhamento dos pacientes que buscavam algum tipo de serviço de saúde, bem como na descoberta de novos sintomas presentes em vítimas do coronavírus, sejam esses mais frequentes ou raros.

[1] https://bit.ly/sintomatic

[2] https://allenai.github.io/scispacy/

Devido à possível mutação do vírus e consequente aparecimento de novas ocorrências de sintomas, como foi o caso da anosmia, tornando-se frequente após um certo período da pandemia em pacientes positivo para COVID19, este modelo proporcionou grandes ganhos no entendimento da doença pela sua capacidade de reconhecer novos padrões.

O Sintomatic é uma rede neural que processa textos em Linguagem Natural, capaz de identificar sintomas a partir de mensagens trocadas entre o *chatbot* e o paciente, bem como reconhecer novos padrões da doença anteriormente inexistente ou despercebidos. Esse tipo de inteligência pode ser perfeitamente treinado para reconhecer e capturar outras classes de palavras em qualquer contexto desejado.

A detecção de sintomas no idioma português foi um desafio, pois, até o momento, não havia de forma pública nenhum modelo capaz de realizar essa tarefa, de acordo com o conhecimento dos autores. O robô desenvolvido foi treinado através de um processo de aprendizado conhecido como *Transfer Learning* [Pan and Yang 2009], ou em português, aprendizado por transferência.

A inovação tecnológica promovida pelo Sintomatic é um modelo neural pioneiro no reconhecimento de sintomas em português, principalmente porque a língua portuguesa carece de modelos *NER*.

A técnica de aprendizagem por transferência utiliza o conhecimento adquirido ao resolver um problema e aplicá-lo em outro problema diferente, porém relacionado, permitindo progresso rápido e desempenho aprimorado ao modelar a segunda tarefa. Em outras palavras, a transferência de aprendizado é a melhoria do aprendizado em uma nova tarefa através da transferência de conhecimento de uma tarefa relacionada que já foi aprendida.

Para treinar o Sintomatic foi utilizado o *scispaCy*, um pacote *Python* que contém modelos de *spaCy* [Honnibal and Montani 2017] para processar textos biomédicos, científicos ou clínicos.

Em particular, há um tokenizador personalizado que adiciona regras de tokenização baseando-se em regras do *spaCy*, um etiquetador *POS* e analisador sintático treinado em dados biomédicos e um modelo de detecção de extensão de entidade. Separadamente, também existem modelos *NER* para tarefas mais específicas.

Para este trabalho o modelo utilizado foi o *en_ner_bc5cdr_md* do *SciSpacy*, em um processo de *transfer leaning* para treinar um novo modelo de reconhecimento e captura de sintomas em português.

A primeira etapa do processo de treino do rastreador foi traduzir os textos que inicialmente estavam em língua portuguesa para o idioma inglês. Em seguida, inserir como parâmetro de entrada cada texto (em inglês) ao modelo do *scispacy*, analisar o resultado gerado por este modelo, e logo após traduzir os sintomas capturados pelo modelo do *scispacy* em inglês para português.

O conjunto de treinamento para o Sintomatic (novo modelo em português), é composto do texto original e os sintomas capturados pelo modelo do *scispacy* em português. Esse processo foi executado de forma contínua até que a função de erro da rede se estabilizasse.

Ao final, foi possível atingir para o Sintomatic, *F1-score* de 85.66, o que é competitivo se comparado ao modelo em inglês, que tem *F1-score* igual a 85.02.

A Figura 1 ilustra as etapas desse processo.

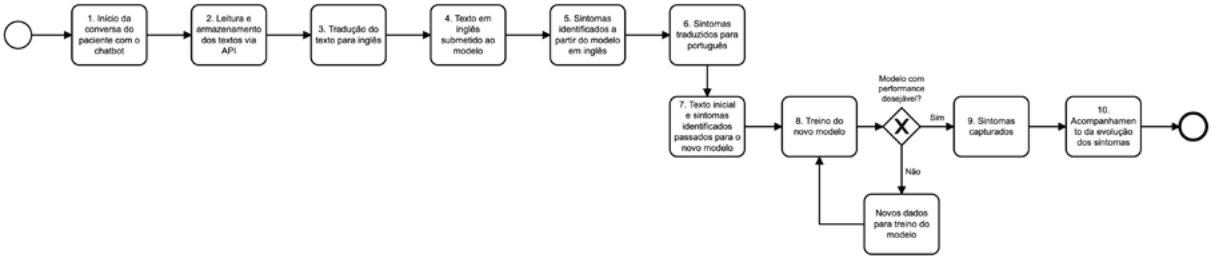


Figura 1. Fluxo dos dados

Nas etapas de translação dos textos foi utilizada a rede tradução do *Google*. Atualmente essas redes de tradução apresentam resultados muito fiéis ao esperado, tornando os ruídos insignificantes quando analisados no contexto deste trabalho.

Um diferencial do Sintomatic é a ausência da necessidade de classificação manual realizada por um humano para reconhecimento de entidades. Em um cenário onde havia vasta quantidade de dados e pouco tempo para processar essas informações, o ganho com a otimização dessa etapa de treino foi crucial no apoio a tomada de decisão.

Outro quesito inovador promovido pelo robô de captura, foi a capacidade de aprender a reconhecer comportamentos de saúde mental.

A partir desta contribuição profissionais de saúde e respectivos órgãos competentes, podem valer-se de tal dado para elaborar e promover políticas públicas com o propósito de assistir a essas pessoas que são acometidas por problemas que ultrapassam a esfera epidemiológica.

Atualmente, o Sintomatic é utilizado na plataforma de Tele Atendimento do Estado do Ceará, onde desempenha papel pioneiro na área da saúde.

3. Cenários de Demonstração

Em um momento de grandes transformações ocasionados pela pandemia por COVID19, surgiu a necessidade de escalar um serviço de saúde de forma rápida e segura, tanto para pacientes como para profissionais de saúde.

A partir desse cenário, foi disponibilizado para a população do Estado do Ceará um serviço de Tele Atendimento gratuito, onde o paciente inicialmente trocava mensagens com um robô, era triado de acordo com seus sintomas e, posteriormente, encaminhado para uma consulta com um profissional de saúde.

Todo esse ciclo de integração com o paciente registrado por meio de textos é passado ao modelo Sintomatic para que este possa detectar sintomas em todas as etapas do atendimento.

A Figura 2 exemplifica parte de uma conversa com um paciente anônimo:

Para o acompanhamento dos dados capturados pelo robô Sintomatic e monitoramento das demais informações sobre a pandemia, foi desenvolvido o Boletim Digital

Figura 2. Trecho da conversa entre o paciente e o *chatbot*

COVID-19 do Ceará, solução tecnológica construída por cientistas de dados onde é feito todo o processo de mineração do dado bruto até sua exposição em painéis gráficos acompanhados de textos explicativos à respeito de cada uma das análises abaixo:

- número de pacientes atendidos;
- sintomas mais frequentes e raros;
- evolução dos sintomas por semana epidemiológica;
- sintomas ao longo do tempo.

A Figura 3 ilustrada na próxima seção desta demonstração expõe a evolução dos sintomas em uma série temporal. Através dessa imagem pode-se identificar a detecção de um novo sintoma no dia oito de maio, perda de olfato. Este sintoma apareceu e tornou-se bastante característico da COVID19 após um certo período de tempo.

Ainda sobre a série temporal, é possível visualizar que a frequência de cada sintoma é sazonal durante o período analisado. Comportamentos de saúde mental, como ansiedade também podem ser observados dentre os sintomas desse gráfico.

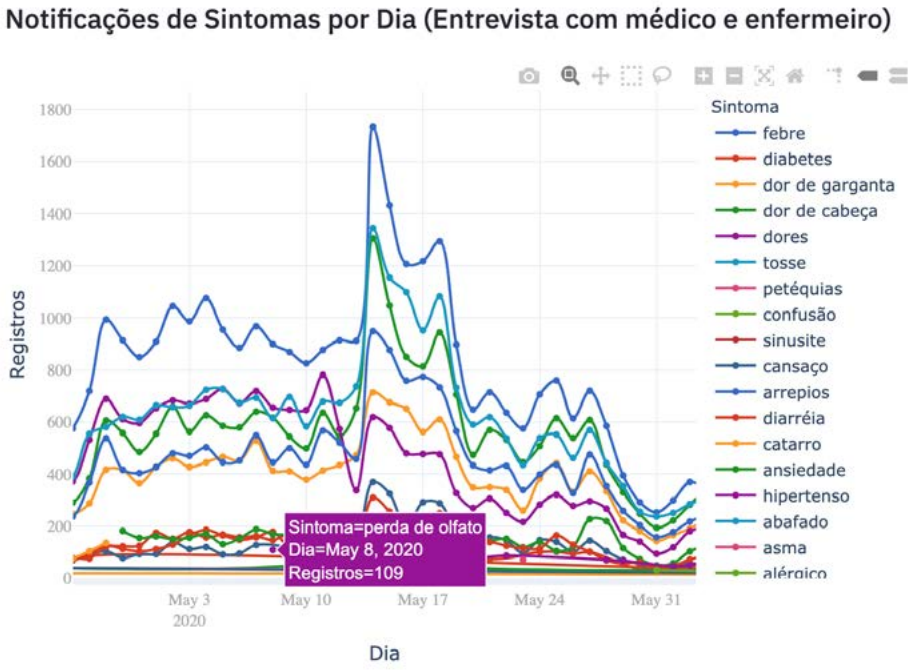
Todo o processo de desenvolvimento e treino do Sintomatic foi realizado em um curto intervalo de tempo e com poucos recursos de mão de obra, pois, dada a atipicidade e urgência da situação, este foi um projeto que precisou ser desenvolvido sem a tradicional estruturação prévia.

Demais estudos também foram realizados com a finalidade de subsidiar a tomada de decisão guiada por dados. Todos os analíticos estão disponíveis no Boletim Digital COVID19 no Ceará.

4. Conclusão

Nessa demonstração, é proposto o uso de um modelo de aprendizado de máquinas para identificar sintomas e comportamentos mentais alterados na população do Estado do Ceará durante a crise causada por coronavírus, chamado Sintomatic.

Ao longo de quatro meses de pandemia no Estado, diversos sintomas foram visualizados. É possível verificar na Figura 3 que a manifestação desses sintomas variam consideravelmente em relação ao tempo, assim como novas ocorrências também foram identificadas ao longo do período analisado.



Fonte: Ceará Tele Atendimento

Figura 3. Sintomas ao longo dos dias

Além de reconhecer novos padrões de sintomas causados por SARS-COV-2, uma das principais contribuições deste trabalho é identificar comportamentos psicológicos alterados, como: ansiedade, angústia e tristeza em pacientes positivos ou não para COVID19.

Diante dessa informação, a Secretaria de Saúde do Estado do Ceará, ou qualquer outro órgão que faça uso dessa tecnologia, pode desenvolver políticas com o propósito de acompanhar essas pessoas em um quadro clínico que acomete não apenas sua saúde fisiológico, como também emocional.

Referências

da Silva, T. L. C., Magalhães, R. P., de Macêdo, J. A., Araújo, D., Araújo, N., de Melo, V., Olímpio, P., Rego, P. A., and Neto, A. V. L. (2019). Improving named entity recognition using deep learning with human in the loop. In *EDBT*, pages 594–597.

Honnibal, M. and Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1).

Neumann, M., King, D., Beltagy, I., and Ammar, W. (2019). Scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*.

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Tarcar, A. K., Tiwari, A., Rao, D., Dhaimodker, V. N., Rebelo, P., and Desai, R. (2020). Healthcare ner models using language model pretraining. In *HSDM@ WSDM*, pages 12–18.

Desenvolvimento e Implementação do Painel COVID-19 para Municípios da Região Norte Fluminense

André Branco[1], Isabelle Thomaz[1], Janaína Gomide[1], Laura Santana[1], Matheus Ferreira[1], Carlos Bazilio[2], Leila Weitzel[2], Leonardo Carvalho[2]

[1]Engenharia – Universidade Federal do Rio de Janeiro (UFRJ)
Macaé – RJ – Brazil

[2]Instituto de Ciência e Tecnologia - Universidade Federal Fluminense (UFF)
Rio das Ostras, RJ – Brazil
{janainagomide,lauraemmanuella,matheusferreira.ufrj}@gmail.com,
{leila_weitzel,carlosbazilio,leonardooc}@id.uff.br

Abstract. *The COVID-19 pandemic highlighted the importance of data visualization with information dashboards, since they assist the decision-making process. This paper presents the COVID-19 Dashboard for some cities from Northern Rio de Janeiro[1]. This dashboard consolidates the COVID-19 data provided on social media and epidemiological bulletins by the prefectures. The visualizations were built for public consultation, from municipal managers, health professionals, researchers and citizens. On the dashboard, information can be viewed separately by municipality, or comparatively between them. The source code is available at GitHub, and can be used by other municipalities and regions that may be interested in developing their own dashboards.*

Resumo. *A pandemia da COVID-19 evidenciou a importância da visualização de dados a partir de painéis de informação, visto que auxiliam o processo de tomada de decisão. Neste artigo, apresenta-se o Painel COVID-19 para cidades do Norte Fluminense[1]. Este painel consiste na consolidação dos dados disponibilizados pelas prefeituras nas redes sociais e nos boletins epidemiológicos. As visualizações foram construídas para consulta pública, de gestores municipais, profissionais de saúde, pesquisadores e cidadãos. No dashboard as informações podem ser vistas separadamente por município, ou de forma comparativa entre eles. O código fonte está disponível no GitHub, e pode ser utilizado por outras prefeituras e regiões que desejam desenvolver seus próprios painéis.*

1. Introdução

O processo de análise de dados tornou-se mais relevante nos dias atuais em virtude do grande volume de dados disponíveis. Segundo [Amaral 2016], o ciclo de vida dos dados envolve a produção, armazenamento, transformação, análise e descarte. Uma consequente busca por conhecimentos contidos nesses dados permite uma melhor tomada de decisão por parte dos gestores, sejam públicos ou privados. Neste contexto, a visualização de dados a partir de painéis de informação (*dashboards*) desempenha um papel importante, pois permite aos gestores obterem *insights* e tomarem decisões baseadas em um panorama mais amplo.

[1]Link: `https://painelcovid19.macae.ufrj.br/`

A disponibilidade de informações precisas de dados clínicos, epidemiológicos e laboratoriais de uma epidemia faz-se necessária para orientar a saúde pública na tomada de decisão. É importante entender a transmissibilidade, o risco de propagação geográfica, rotas de transmissão e fatores de risco de infecção. Desta forma, o painel de informações de saúde fornece a linha de base para modelagem epidemiológica informando o planejamento da resposta aos esforços de contenção para reduzir o ônus da doença. Além disso, informações detalhadas fornecidas em tempo real são cruciais para decidir onde priorizar a vigilância [Sarrafzadeh 2020].

Os painéis de informações de saúde tornaram-se um recurso e estratégia comum para melhorar a leitura dos dados desse setor, visto que são coleções de visualizações de indicadores relevantes para o gerenciamento e tomadas de decisão [Michelsen et al. 2015]. Há algumas iniciativas que podem ser citadas, como o sistema InfoGripe[2], que realiza o acompanhamento de casos reportados de síndrome respiratória aguda grave. O trabalho apresentado em [Organization et al. 2018] contém um guia para painéis de sistemas de alerta para surtos de dengue. Além disso, em [Silva et al. 2011] os autores propõem uma visualização do acompanhamento da situação da dengue no Brasil.

Devido à pandemia da COVID-19, diversos painéis de informação foram desenvolvidos. Alguns países, estados e regiões criaram seus próprios painéis COVID-19. Há painéis de abrangência nacional como o Brasil.io[3] com dados compilados a partir de boletins epidemiológicos das Secretarias Estaduais de Saúde (SES), o GeoCovid-19[4] apresenta, além dos dados da COVID-19, a taxa de isolamento e as projeções de cenários futuros. Além desses, há o site oficial do Ministério da Saúde para comunicação da situação epidemiológica no Brasil[5] e painéis com dados mais específicos, como o lançado pelo Governo do Estado do Rio de Janeiro[6] que contém números de internações e uso das unidades de terapia intensivas (UTIs) e o painel oficial da cidade do Rio de Janeiro, o Rio COVID-19[7], que apresenta dados a nível de bairros.

Outra forma de divulgação da situação da COVID-19 é por meio da publicação dos dados nas redes sociais ou em boletins oficiais das prefeituras. É disponibilizado aos cidadãos o número de casos da doença no município, número de pessoas recuperadas e número de óbitos. A visualização dessas informações ao longo do tempo, georreferenciadas e de forma comparativa nem sempre é possível, visto que nem todas as prefeituras possuem um painel de informação dessa doença.

Neste artigo, apresenta-se o Painel COVID-19 para cidades do Norte Fluminense. Este painel consiste na consolidação dos dados sobre COVID-19 disponibilizados nas redes sociais e boletins epidemiológicos das prefeituras das cidades de Campos dos Goytacazes, Carapebus, Conceição de Macabu, Macaé, Quissamã, Rio das Ostras, São Fidélis, São Francisco do Itabapoana e São João da Barra. É possível acessar as informações de forma separada para cada município ou comparativa. A implementação dessa aplicação

[2]http://info.gripe.fiocruz.br/help
[3]https://brasil.io/home/
[4]http://covid.mapbiomas.org/
[5]https://covid.saude.gov.br/
[6]http://painel.saude.rj.gov.br/monitoramento/covid19.html
[7]https://experience.arcgis.com/experience/38efc69787a346959c931568bd9e2cc4

foi realizada utilizando ferramentas gratuitas e o código está disponível no GitHub[8].

Esse trabalho traz benefícios diretos para a comunidade dos municípios envolvidos através do fornecimento de informações e conteúdos que poderão ser utilizados tanto pelos gestores em saúde pública para subsidiar o planejamento, execução e avaliação das ações de combate a pandemia de COVID-19 no seu território, quanto por estudantes e pesquisadores de diferentes áreas com interesse no tema. Além disso, o código da ferramenta é aberto e pode ser utilizado por outras prefeituras e regiões para disponibilizar seus painéis de informação seja da COVID-19 ou de outra epidemia que desejam monitorar.

2. Metodologia

A metodologia adotada para implementar a aplicação envolve as seguintes etapas: (1) Coleta de dados; (2) Armazenamento e transformação dos dados e (3) Visualização.

2.1. Coleta de dados

Os dados que compõem o Painel COVID-19 para Região Norte Fluminense são: informações sobre a doença (número de casos confirmados, número de óbitos, número de casos recuperados e indicadores epidemiológicos) e o índice de isolamento social.

O levantamento sobre a doença é realizado diariamente pelos próprios municípios e disponibilizados no Painel Coronavírus do Ministério da Saúde[5]. Além disso, são coletadas informações dos boletins e informes publicados nos sites e nas redes sociais das prefeituras. Para calcular os indicadores epidemiológicos, as taxas de incidência e mortalidade, é necessário saber dados da demografia de cada região. Estes dados estão disponíveis no sítio do IBGE[9] e referem-se à população estimada para 2019.

As informações sobre o índice isolamento social foram gentilmente cedidas pela empresa InLoco[10] através de convênio. O índice varia de 0 a 1. O cálculo envolve dados de GPS de aproximadamente 60 milhões de celulares, o que possibilita o cálculo da movimentação média diária dos indivíduos. Esse índice está disponível para a maioria das cidades Brasileiras.

2.2. Armazenamento e Transformação

O armazenamento dos dados coletados é feito em planilhas (arquivos .xlsx). Há uma planilha para os dados da COVID-19 e outra para os dados do isolamento. Todas as planilhas ficam armazenadas no Google Drive, facilitando o acesso pela aplicação do Painel COVID-19. O *download* dos arquivos do Google Drive para a execução dos *scripts* é feito utilizando a API do Google Drive[11]. As planilhas são lidas e transformadas por meio da biblioteca Pandas[12]. Ao fazer o *download* das planilhas, o *script* mantém os arquivos antigos como *backup* evitando que erros nos novos arquivos impeçam a leitura e tratamento dos dados, comprometendo a visualização. Além disso, o *script* configura o envio de mensagens de erro por *e-mail* para os responsáveis pela manutenção do painel, conferindo maior eficácia e rapidez na resolução de exceções.

[8] https://github.com/gtcovidcomp/painel-covid19
[9] https://www.ibge.gov.br/estatisticas/sociais/populacao/
[10] www.inloco.com.br
[11] https://developers.google.com/drive/api/v3/quickstart/python
[12] https://pandas.pydata.org/

O programa do Unix cron[13] foi utilizado para agendar a execução do *script* que faz a atualização automática (de 4 em 4 horas) das visualizações do painel a partir da adição de informações nas planilhas de dados.

2.3. Visualização

Os sistemas de visualização geralmente são projetados para desenvolver atividades cognitivas de alto nível, como a compreensão de fenômenos específicos, a descoberta de *insights* sobre um problema e tomar uma decisão diante de dados complexos ou massivos. Um procedimento clássico de implementação visual é processar e filtrar os dados, transformá-los em uma forma visual e depois renderizá-los [Few 2013, O'Donoghue et al. 2018].

Em seu estudo, [Heer and Bostock 2010] pesquisaram como os detalhes da percepção humana afetam a capacidade de decifrar as exibições gráficas de dados. O estudo buscava descobrir quais tipos de gráficos são compatíveis com a capacidade humana de interpretação. Seus experimentos mostraram que as pessoas são melhores na leitura de gráficos com base no comprimento de barras ou linhas, como em um gráfico de barras padrão. Verificaram que essas visualizações são a melhor opção quando é importante discernir com precisão pequenas diferenças entre os valores. Os gráficos de barras são mais adaptados quando se estiver visualizando contagens ou proporções. Por outro lado, os gráficos de pizza podem ser mais "atraentes" do que os gráficos de barras, pois são fáceis de preencher com cores e de serem elaborados. Todavia não são uma estratégia eficaz para visualizar dados contínuos, são aceitáveis apenas em contextos limitados.

Baseados no contexto acima, os gráficos selecionados foram os seguintes: gráficos do tipo barra vertical (colunas) associados a gráficos de linha para apresentar o acumulado de casos confirmados, recuperados e óbitos ao longo do tempo; gráficos de linha para apresentar a taxa de isolamento e a comparação de casos confirmados e óbitos entre as cidades ao longo do tempo; e um mapa de bolhas para apresentar um comparativo visual entre os casos confirmados, recuperados e óbitos das cidades do Norte Fluminense. O painel possui um filtro para seleção da cidade, permitindo a visualização dos gráficos apenas da cidade escolhida.

A aplicação para *web* é implementada utilizando Dash[14]. O Dash é um *framework* Python gratuito e *open-source* para criar aplicações *web* e possibilitar a geração de relatórios e *dashboards* para análise de dados e visualizações interativas. Os gráficos e mapas foram implementados utilizando a biblioteca Plotly[15] para Python.

Todos os gráficos e o mapa são objetos da classe *dash_core_components.Graph*, que recebem como parâmetro um objeto da classe *plotly.graph_objects.Figure* que determina o *layout* e dados dos gráficos. O mapa de bolhas foi gerado com a função *plotly.express.scatter_mapbox*, com o parâmetro *animation_frame* igual ao nome da coluna de data, produzindo assim uma animação ilustrando o crescimento dos casos no decorrer dos dias. A formatação e estilo do mapa foram definidos na ferramenta Mapbox Studio[16] e integrados ao painel por meio da API da ferramenta. Em adição aos dados de casos,

[13] `https://e-tinet.com/linux/crontab/`
[14] `http://dash.plotly.com/`
[15] `https://plotly.com/python/`
[16] `https://studio.mapbox.com/`

óbitos e recuperados acumulados por dia, cada cidade foi associada com suas latitude e longitude correspondentes, necessárias para posicionar as bolhas referentes aos dados no mapa.

3. Design do Painel

A tela principal do Painel COVID-19 é apresentada na Figura 1. No cabeçalho estão as logos das instituições, a data da última atualização e o *link* para a página Sobre. Logo abaixo é possível selecionar o município para o qual as informações são exibidas (1). Nessa imagem tem a visualização para o município de Macaé. Nesse painel é possível ver (2) o número de casos confirmados, número de óbitos, número de casos recuperados, taxa de incidência, taxa de letalidade e taxa de mortalidade. Além desses números, é possível visualizar também, (3) gráfico com esses dados ao longo do tempo, (4) o índice de isolamento ao longo do tempo e (5) o mapa com as cidades da região que mostra o número de casos ao longo do tempo. Além disso, há um comparativo entre os municípios do número de casos e óbitos nas cidades ao clicar na aba "Comparação Cidades" (6), e do índice de isolamento ao longo do tempo ao selecionar "Região" no filtro.

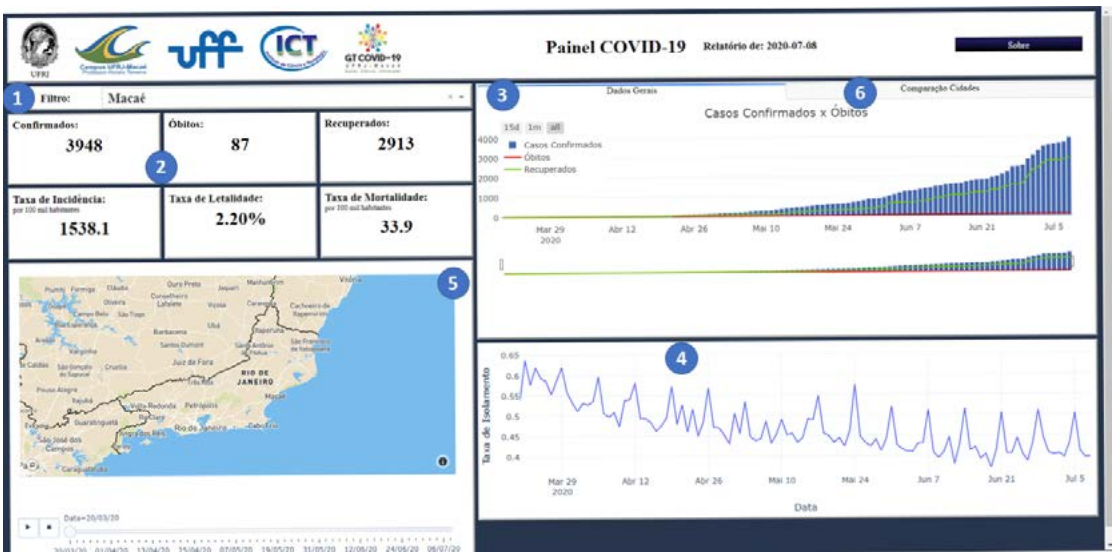


Figura 1. Painel COVID19

As informações do Painel COVID-19, os dados, a equipe responsável e o contato para maiores informações, estão disponíveis na página Sobre.

4. Considerações Finais

Este artigo tem como objetivo apresentar o desenvolvimento e implementação do painel de visualização da COVID-19 dos municípios Norte Fluminense. O painel apresenta dados sobre o número de casos confirmados, casos recuperados e óbitos, indicadores epidemiológicos e a taxa de isolamento da população. As visualizações disponíveis apresentam os dados ao longo do tempo e um mapa com as cidades da região. Há a possibilidade também de comparar os dados entre os municípios.

A aplicação foi desenvolvida utilizando a linguagem Python e as bibliotecas Dash e Plotly. O Painel COVID-19 pode ser acessado pelo link

www.painelcovid19.macae.ufrj.br. O código fonte desse painel é aberto e pode ser consultado no GitHub[17]. Dessa forma, o código encontra-se disponível para download e pode ser reaproveitado por prefeituras e regiões que desejam desenvolver seus próprios painéis.

Esta aplicação traz como contribuição a organização das informações para auxiliar os gestores na tomada de decisão, bem como a difusão da informação e a visualização com maior clareza das tendências para a população. Devido à importância e à urgência do tema, os pesquisadores das instituições parceiras uniram-se para que em tempo hábil pudessem disponibilizar a aplicação. Novas visualizações estão sendo desenvolvidas, como incidência da doença por sexo, faixa etária, bairro e análise dos sintomas mais frequentes para os municípios que disponibilizarem essas informações.

Referências

Amaral, F. (2016). *Introdução à ciência de dados: mineração de dados e Big Data*. Alta Books, 1st edition.

Few, S. (2013). *Information dashboard design: displaying data for at-a-glance monitoring*. Analytics Press, Burlingame, CA, second edition edition. 00200 OCLC: 856809336.

Heer, J. and Bostock, M. (2010). Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 203–212, New York, NY, USA. Association for Computing Machinery.

Michelsen, T., Grawunder, M., Geesen, D., and Appelrath, H.-J. (2015). Demo: Dynamic generation of adaptive real-time dashboards for continuous data stream processing. In Castellanos, M., Dayal, U., Pedersen, T. B., and Tatbul, N., editors, *Enabling Real-Time Business Intelligence*, pages 171–174, Berlin, Heidelberg. Springer Berlin Heidelberg.

O'Donoghue, S. I., Baldi, B. F., Clark, S. J., Darling, A. E., Hogan, J. M., Kaur, S., Maier-Hein, L., McCarthy, D. J., Moore, W. J., Stenau, E., Swedlow, J. R., Vuong, J., and Procter, J. B. (2018). Visualization of biomedical data. *Annual Review of Biomedical Data Science*, 1(1):275–304.

Organization, W. H., for Research, U. B. S. P., and in Tropical Diseases, T. (2018). *Operational guide using the web-based dashboard: Early Warning and Response System (EWARS) for dengue outbreaks*. World Health Organization.

Sarrafzadeh, M. (2020). Olivia health analytic platform. In *Proceedings of Deep Learning for Wellbeing Applications Leveraging Mobile Devices and Edge Computing*, HealthDL'20, page 9, New York, NY, USA. Association for Computing Machinery.

Silva, I., Gomide, J., Barbosa, G., Filho, W., Veloso, A., Meira Jr., W., and Ferreira, R. (2011). Observatório da dengue: Surveillance based on twitter sentiment stream analysis. In *Proceedings of XXVI Simpósio Brasileiro de Banco de Dados - Demo Papers*. SBC.

[17]Link GitHub: `https://github.com/gtcovidcomp/painel-covid19`

Ferramenta brModelo: Quinze Anos!

Ronaldo dos Santos Mello[1], Carlos Henrique Cândido[2], Milton Bittencourt S. Neto[3]

[1]Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC

[2]Universidade Federal do Mato Grosso (UFMT) – Cuiabá, MT

[3]Mercado Livre – Florianópolis, SC

r.mello@ufsc.br,chcandido@gmail.com,miltonbst@gmail.com

Abstract. *A ferramenta brModelo foi desenvolvida pelo Grupo de Banco de Dados da UFSC em 2005 com o intuito de ser uma ferramenta gratuita para apoiar o ensino de projeto de bancos de dados relacionais. Seus principais diferenciais em relação a ferramentas similares são o suporte a todas as três etapas clássicas de projeto de banco de dados, a interação com o projetista durante a execução da etapa de modelagem lógica e o suporte a todos os conceitos do modelo EER descritos na principal literatura nacional sobre projeto de banco de dados. A aceitação da brModelo tem sido grande nesses seus quinze anos de existência, o que motivou a geração de diversas outras versões. Esse artigo apresenta um pouco da história da brModelo, incluindo suas versões atualmente disponíveis e suas funcionalidades.*

1. Introdução

O projeto ou modelagem de um Banco de Dados (BD) é um processo fundamental no desenvolvimento e manutenção de uma infraestrutura computacional, uma vez que seu resultado é o esquema de um repositório de dados que pode ser acessado por um ou mais sistemas de uma organização, sendo estes dados vitais para o adequado funcionamento desta organização [Batini et al. 1992]. Um projeto de BD visa garantir uma abstração adequada dos dados do domínio, bem como garantir armazenamento e acesso eficientes a estes dados. A não consideração de uma metodologia de projeto de BD pode gerar diversos problemas, como compreensão parcial ou incorreta do domínio do problema, redundância de dados e baixo desempenho de acesso.

A *brModelo* é uma ferramenta de apoio ao projeto de BDs relacionais desenvolvida pelo *Grupo de BD da UFSC (GBD/UFSC)*[1]. Passados quinze anos da sua criação e da sua divulgação na comunidade nacional de BD, verifica-se hoje o seu amplo uso em disciplinas de BD em nível de graduação e pós-graduação, bem como em cursos voltados ao projeto de BD Brasil afora. Um exemplo disso são os diversos tutoriais produzidos e disponibilizados por professores e profissionais que a utilizam[2]. O principal motivo para a popularidade da *brModelo* ao longo de todo esse tempo são os seus principais diferenciais em relação a ferramentas com propósito similar: *(i)* suporte a todas as etapas tradicionais de projeto de um BD: conceitual, lógica e física; *(ii)* auxílio na tomada de decisões durante a geração da modelagem lógica; *(iii)* suporte a todos os conceitos do

[1]http://lisa.inf.ufsc.br/wiki/index.php/Main

[2]https://www.youtube.com/results?search_query=brmodelo

modelo entidade-relacionamento estendido (modelo EER) conforme proposto pela principal literatura nacional sobre projeto de BD [Heuser 2008].

A ferramenta foi aprimorada ao longo do tempo, gerando diversas versões cujas principais funcionalidades são descritas na Seção 3. Além desta seção, a Seção 2 conta um pouco da história da *brModelo* ao longo dos seus quinze anos de existência, a Seção 4 comenta ferramentas relacionadas e a Seção 5 é dedicada à conclusão.

2. Um Pouco de História

O projeto de um BD é um conteúdo obrigatório em disciplinas da área de BD devido a sua grande relevância. Nesse contexto, uma das principais referências bibliográficas a respeito de projeto de BD relacional em nível nacional é o livro do professor *Carlos A. Heuser* intitulado *Projeto de BD*, cuja primeira edição ocorreu em 2001 [Heuser 2001]. Entretanto, alguns anos após o lançamento deste livro, percebeu-se a ausência não apenas de uma ferramenta de apoio ao ensino de modelagem de dados que adotasse a notação utilizada por essa referência, como também uma ferramenta que contemplasse todas as três etapas tradicionais de um projeto de BD relacional. Essas constatações motivaram, em 2005, o desenvolvimento da primeira versão da ferramenta *brModelo* pelo GBD/UFSC a partir de um trabalho em nível de especialização *latu sensu* [Cândido 2005].

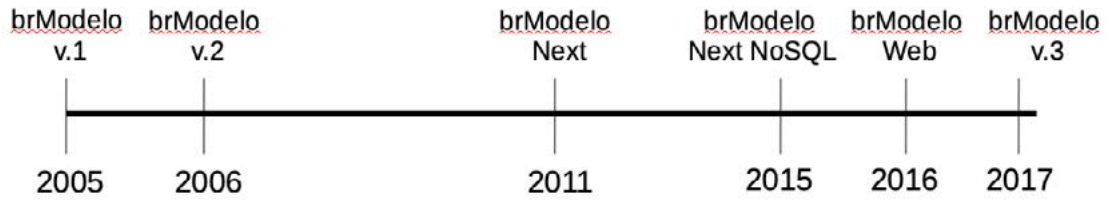


Figure 1. *Timeline* da ferramenta brModelo

A Figura 1 mostra o *timeline* com os principais fatos relacionados à história da *brModelo*. Sua primeira versão foi desenvolvida na linguagem de programação Delphi e estava restrita à execução no sistema operacional *Microsoft Windows*. A segunda versão da ferramenta, lançada no ano seguinte, mantinha essa restrição de execução apenas na plataforma *Windows*, porém, corrigiu alguns erros de usabilidade e alguns erros na geração do esquema lógico a partir de um esquema EER.

Mesmo com a sua utilização cada vez maior em cursos de graduação na área da Computação, a *brModelo* recebia diversos apelos para se tornar uma ferramenta multiplataforma. Esses apelos foram atendidos em 2011 com o surgimento da versão *brModeloNext*. A *brModeloNext*, apresentada na Sessão de Demos do SBBD 2011, foi totalmente reimplementada na linguagem Java e pode agora ser executada em qualquer computador com uma máquina virtual Java instalada [Menna et al. 2011]. Por ter sido reimplementada totalmente por um grupo diferente de desenvolvedores, a *brModeloNext* apresenta uma interface com o usuário mais moderna e amigável. Devido a esse novo visual, essa versão foi considerada uma ”nova geração” da ferramenta, justificando a inclusão do termo *”Next”* ao seu nome.

Em 2015, o protótipo de uma nova versão da *brModeloNext*, denominada *brModeloNext NoSQL*, foi disponibilizado. Ele oferece a possibilidade de gerar um esquema lógico denominado *esquema de agregados*, que atua como uma abstração canônica para

três modelos de dados NoSQL: chave-valor, orientado a colunas e orientado a documentos. O esquema de dados baseado em agregados foi definido por uma literatura clássica sobre BDs NoSQL [Sadalage and Fowler 2012], e sua notação gráfica foi proposta pelo GBD/UFSC [Lima and Mello 2015]. A *brModeloNext NoSQL* encontra-se ainda em desenvolvimento, apesar de uma versão de testes já estar disponível (ver Seção 5).

Todas as versões da *brModelo* até 2015 se caracterizavam por ser ferramentas *desktop*, ou seja, só podiam ser baixadas e executadas localmente. Isso mudou em 2016 com o surgimento da *brModeloWeb*, a primeira versão da ferramenta acessível através da Internet [Neto 2016]. Apesar de apresentar menos funcionalidades que as versões *desktop*, ela facilitou bastante o acesso à ferramenta por qualquer pessoa, beneficiando principalmente aulas de modelagem de dados em laboratórios de Informática. A *brModeloWeb* foi apresentada à comunidade de BD na forma de uma oficina ministrada na Escola Regional de Banco de Dados (ERBD) de 2017.

Nesse mesmo evento (ERBD 2017) também foi lançada a terceira versão da linhagem *brModelo*: a ferramenta *brModelo v.3* [Cândido and Mello 2017]. Esta é a atual versão *desktop* da ferramenta, que foi também reimplementada em Java para se tornar multiplataforma e disponibiliza editores para outras notações diagramáticas úteis no desenvolvimento de software, como diagramas de atividades e de fluxo de dados.

3. Principais Funcionalidades

Todas as ramificações e versões da *brModelo* apresentadas na seção anterior compartilham as seguintes funcionalidades: *(i)* suporte às três etapas tradicionais do projeto de um BD relacional; *(ii)* a geração da modelagem lógica é guiada pelo usuário projetista; *(iii)* interface gráfica com o projetista intuitiva e rica em opções de menu e ícones representando conceitos de modelagem que podem ser manipulados no estilo *drag-and-drop*.

Essas funcionalidades podem ser vistas na Figura 2 para a ferramenta *brModelo v.3*. À esquerda é possivel ver a criação de uma modelagem conceitual utilizando a notação EER de Heuser (Figura 2 (a)). Os ícones à direita na Figura 2 (a) representam os conceitos do modelo EER, que podem ser selecionados e arrastados para a área de trabalho central. Já na Figura 2 (b) vê-se um exemplo de interação do projetista durante a geração da modelagem lógica correspondente à modelagem conceitual da Figura 2 (a). Neste caso, toda vez que existe mais de uma opção de mapeamento de um conceito da modelagem conceitual para um esquema relacional, a ferramenta apresenta essas opções para o projetista selecionar uma delas. As opções de menu e os ícones na parte superior da interface oferecem as operações mais comuns de manipulação (salvar, criar nova modelagem, etc), bem como a passagem para uma próxima etapa do projeto do BD.

A ferramenta *brModeloNext*, por sua vez, agregou diversas melhorias em termos de interação com o usuário, sendo a principal delas a possibilidade de se trabalhar com múltiplas janelas, como mostra a Figura 3 (a). Neste exemplo, o projetista visualiza simultaneamente as modelagens conceitual, lógica e física dos dados que está criando, podendo manipular cada uma delas.

A Figura 3 (b) mostra a interface da versão *brModeloNext NoSQL*. Conforme descrito na seção anterior, esta versão permite gerar uma modelagem lógica baseada em agregados a partir de uma modelagem conceitual EER. Um agregado é um esquema de

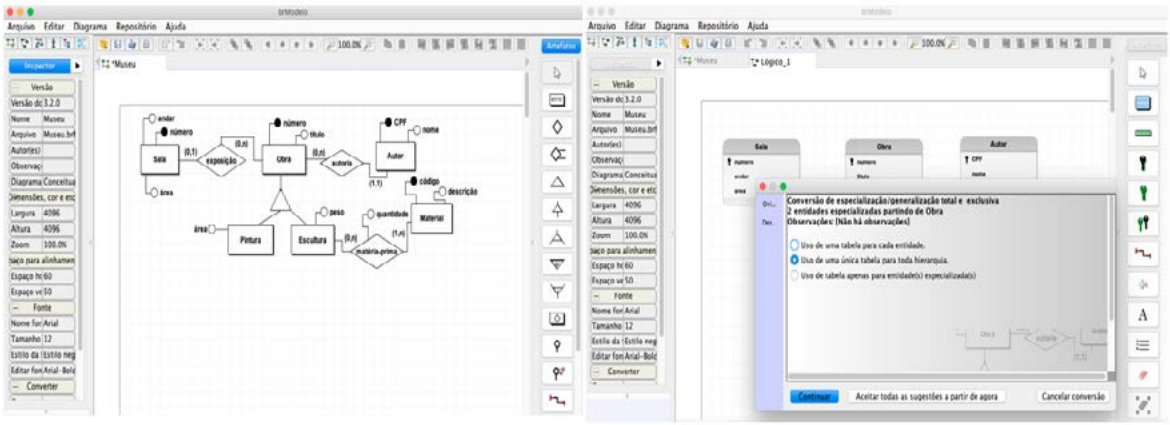


Figure 2. brModelo v.3: modelagem conceitual (a) e modelagem lógica (b)

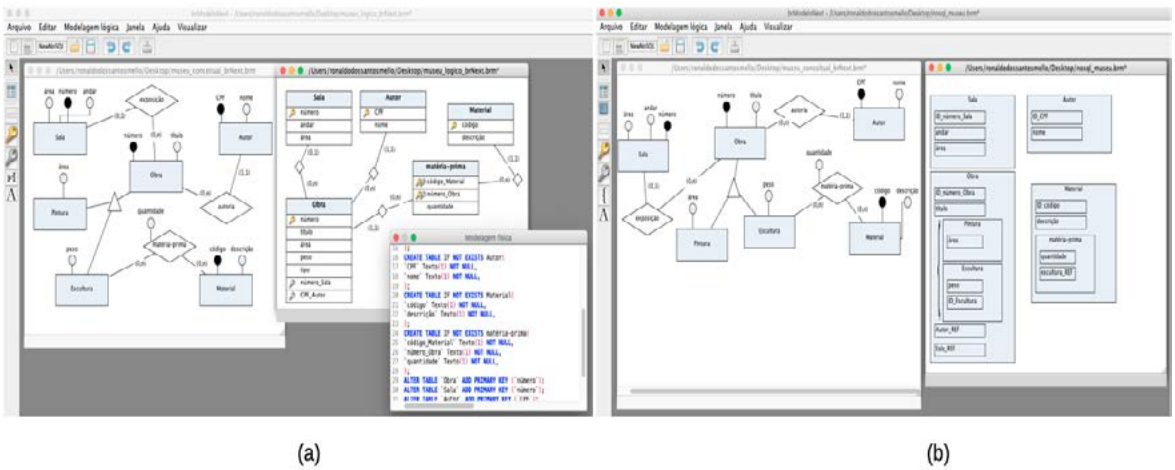


Figure 3. brModeloNext: suporte a múltiplas janelas (a) e a versão NoSQL (b)

um objeto complexo, ou seja, um esquema que pode agregar outros esquemas de objetos, como é o caso do esquema do objeto *Obra*, que encapsula seus atributos e os esquemas dos objetos *Pintura* e *Escultura*.

Por fim, a Figura 4 apresenta algumas telas da *brModeloWeb*. A Figura 4 (a) mostra a tela de entrada da ferramenta na qual é possível criar uma nova conta para acesso e se logar remotamente no servidor da ferramenta. A Figura 4 (b) exibe a área de trabalho do projetista, com as suas modelagens já criadas e salvas na nuvem. Nesta tela também é possível criar novas modelagens conceituais ou lógicas. Já a Figura 4 (c) apresenta parte da interface para modelagem conceitual, que é similar às interfaces das demais versões da *brModelo*.

A versão *brModeloWeb*, por ser mais recente, ainda carece de algumas funcionalidades presentes nas demais versões, como a exportação das modelagens criadas para o computador local do projetista em algum formato de arquivo. Esta e outras operações já estão previstas para a próxima versão.

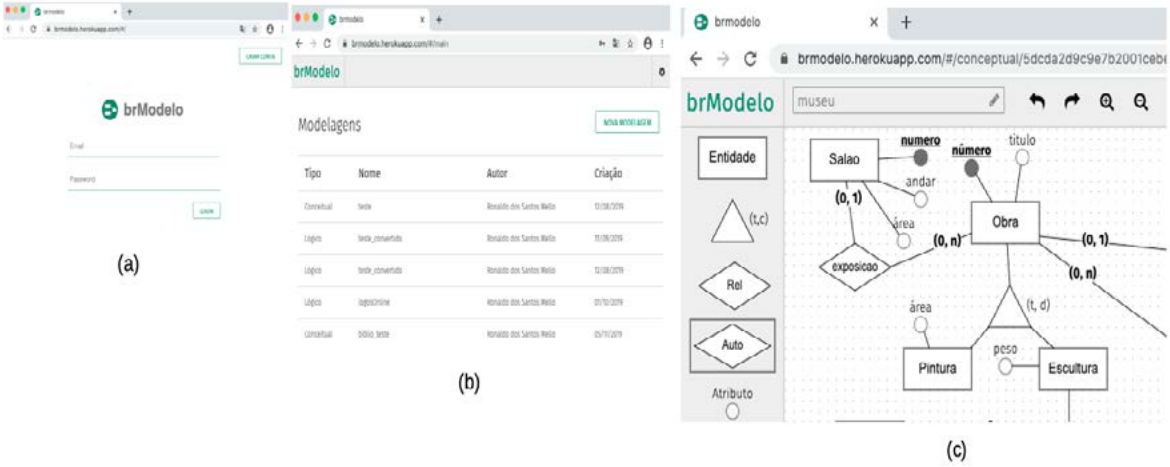


Figure 4. brModeloWeb: interface de entrada (a), interface da área de trabalho (b) e interface para modelagem conceitual (c)

4. Trabalhos Relacionados

Diversas soluções comerciais para apoio ao projeto de BD relacional encontram-se disponíveis, como a *ERwin, DBDesigner, Enterprise Architect, RISE Editor, DeZign, Power Designer, Oracle Designer, ER/Studio, Toad Data Modeler, Open ModelSphere* e *DB-Main*, além de algumas ferramentas acadêmicas, como *TerraER* [Rocha and Terra 2010] e *EERCASE* [Alves et al. 2014]. As principais limitações dessas soluções, se comparadas com a *brModelo*, são a falta de suporte para as três etapas de projeto de BD, a falta de cobertura de todos os conceitos do modelo EER, bem como a utilização de uma notação diferente da notação de Heuser.

Conforme salientado anteriormente, um importante diferencial da *brModelo* é a flexibilidade no mapeamento de esquemas conceituais para esquemas lógicos. Ela executa este mapeamento de forma semiautomática, ou seja, oferecendo a possibilidade de escolha de uma dentre diversas alternativas de conversão de um conceito da modelagem EER. Assim, o usuário tem a liberdade de orientar a conversão para uma estrutura lógica mais adequada ao seu domínio.

5. Conclusão

A iniciativa *brModelo* teve início em 2005 e se propagou ao longo destes quinze anos graças ao esforço conjunto de bolsistas do GBD/UFSC e de colaboradores responsáveis por algumas de suas versões e que continuam se dedicando ao seu desenvolvimento. Um agradecimento especial vai para *Carlos H. Cândido* e *Milton Bittencourt de S. Neto*, responsáveis pelo desenvolvimento das versões *brModelo* e *brModeloWeb*, respectivamente, e que até hoje colaboram com o aprimoramento delas.

O resultado de todo esse esforço é a grande aceitação que essa ferramenta teve e continua tendo, principalmente no ensino de projeto de BD em cursos de graduação em computação e treinamentos de modelagem de dados por todo o Brasil. Uma grande quantidade de vídeoaulas sobre a ferramenta estão à disposição no *YouTube*, com centenas de milhares de visualizações e *downloads*, até onde foi possível investigar. Em 2017, quando foi lançada a *brModelo v.3*, a quantidade de *downloads* da ferramenta já superava quinhentos mil. Essa quantidade deve ter aumentado bastante até os dias

de hoje. Todo esse grande interesse pela *brModelo* não era esperado e isso deixa o GBD/UFSC bastante contente! As principais versões *desktop* da *brModelo* estão disponíveis para *download* na *Wiki* do GBD/UFSC[3]. Já a *brModeloWeb* está acessível em *https://www.brmodeloweb.com*. Ela é hoje um projeto *open-source* e contribuições são bem-vindas em *https://github.com/brmodeloweb/brmodelo-app*.

Diversas melhorias de funcionalidade estão em andamento nas versões da *brModelo*, graças principalmente a sugestões recebidas ao longo do tempo. Exemplos dessas melhorias são a exportação detalhada dos metadados dos projetos criados, o suporte a alguns conceitos do modelo EER presentes em outras notações na literatura, bem como a internacionalização da ferramenta. Vida longa e próspera à *brModelo*!

References

Alves, E., Franco, N., Nascimento, A., and Fidalgo, R. (2014). EERCASE: Uma Ferramenta para Apoiar o Estudo do Projeto Conceitual de Banco de Dados. In *Workshops do III Congresso Brasileiro de Informática na Educação (CBIE)*.

Batini, C., Ceri, S., and Navathe, S. B. (1992). *Conceptual Database Design: An Entity-Relationship Approach*. Benjamin/Cummings.

Cândido, C. H. (2005). *Aprendizagem em Banco de Dados: Implementação de Ferramenta de Modelagem E.R.* Monografia de Especialização. Universidade Federal de Santa Catarina. 44p.

Cândido, C. H. and Mello, R. (2017). Ferramenta de Modelagem de Banco de Dados Relacionais *brModelo v3*. In *XIII Escola Regional de Banco de Dados (ERBD)*.

Heuser, C. A. (2001). *Projeto de Banco de Dados*. Sagra Luzzatto, 4 edition.

Heuser, C. A. (2008). *Projeto de Banco de Dados*. Bookman, 6 edition.

Lima, C. and Mello, R. (2015). A Workload-Driven Logical Design Approach for NoSQL Document Databases. In *XVII International Conference on Information Integration and Web-based Applications & Services (iiWAS)*. ACM.

Menna, O. S., Ramos, L. A., and Mello, R. (2011). *brModeloNext*: a Nova Versão de uma Ferramenta para Modelagem de Bancos de Dados Relacionais. In *VI Sessão de Demos do Simpósio Brasileiro de Banco de Dados (SBBD Demos)*.

Neto, M. B. (2016). *brModeloWeb*: Ferramenta Web para Ensino e Modelagem de Banco de Dados. Trabalho de Conclusão de Curso em Ciência da Computação. Universidade Federal de Santa Catarina. 70p.

Rocha, H. S. C. and Terra, R. (2010). *TerraER*: Uma Ferramenta voltada ao Ensino do Modelo de Entidade-Relacionamento. In *VI Escola Regional de Banco de Dados (ERBD)*.

Sadalage, P. J. and Fowler, M. (2012). *NoSQL Distilled : A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley.

[3]http://lisa.inf.ufsc.br/wiki/index.php/Projects

QualiOSM: Melhorando a Qualidade dos Dados na Ferramenta de Mapeamento Colaborativo OpenStreetMap

Gabriel F. B. de Medeiros[1], Lívia C. Degrossi[2], Maristela Holanda[1],

[1]Departamento de Ciências da Computação – Universidade de Brasília (UnB)
Brasília – Brasil

[2]Fundação Getúlio Vargas (FGV)
São Paulo – Brasil

{gabriel.medeiros93,maristela.holanda,liviadegrossi}@gmail.com

Abstract. *OpenStreetMap (OSM) is a large spatial database in which geographic information is voluntarily contributed by thousands of users. The issue of data quality in collaborative systems is challenging, since users without technical knowledge actively participate in the processes of including, editing and excluding information. In the case of OSM, the attributes of the objects are present in the form of labels called tags, and the process of assigning these tags contributes to improving the attribute completeness, corresponding in this context to an important metric of data quality. In this way, this work proposes the implementation of the QualiOSM tool, which automatically generates a tag adder with the purpose of improving the completeness of address information for OSM objects in Brazil. The tool was tested in three different scenarios: urban environment, rural environment and slum environment.*

Resumo. *O OpenStreetMap (OSM) é um grande banco de dados espaciais em que as informações geográficas são inseridas voluntariamente por milhares de usuários. A questão da qualidade dos dados em sistemas colaborativos é um desafio, uma vez que usuários sem o devido conhecimento técnico participam ativamente dos processos de inclusão, edição e exclusão das informações. No caso do OSM, os atributos dos objetos estão presentes na forma de etiquetas denominadas de tags, sendo que o processo de atribuição dessas tags contribui para a melhoria da completude dos atributos, correspondendo a uma métrica importante da qualidade dos dados. Dessa forma, este trabalho propõe a implementação da ferramenta QualiOSM[1], a qual gera automaticamente um adicionador de tags com o objetivo de melhorar a completude das informações de endereço para objetos do OSM no Brasil. A ferramenta foi testada em três diferentes cenários: ambiente urbano, ambiente rural e ambiente de favela.*

1. Introdução

Um Sistema de Informação Geográfica (SIG) é um sistema computacional que armazena dados espaciais, cujas funções são controladas interativamente por um componente humano com a finalidade de gerar informações geográficas sobre a superfície terrestre

[1]Video de demonstração da ferramenta disponível em: https://1drv.ms/u/s!AvGoFS456yU_shtEQv23VDO0K0t0?e=kQ7rjm [Acesso em agosto de 2020.]

[Tomlinson 2007]. Com o desenvolvimento de novas tecnologias a partir dos anos 2000, surgiram sistemas em que os usuários são capazes de gerar informações geográficas de forma voluntária e, consequentemente, esses sistemas ficaram popularmente conhecidos como Sistemas de Informações Geográficas Voluntárias (SIGV) [Goodchild 2007]. Esses tipos de sistemas demandam uma atenção maior em relação à qualidade das informações, uma vez que usuários sem o devido conhecimento técnico participam ativamente dos processos de inclusão, alteração e exclusão dos dados.

De acordo com a ISO 19157, a qual estabelece os princípios que descrevem o conceito de qualidade dos dados geográficos, a qualidade pode ser definida como o grau em que um conjunto de características atende a um grupo de requisitos preestabelecidos [ISO 2013]. Dessa forma, o conceito de qualidade de dados costuma ser dividido na literatura em diferentes aspectos, os quais foram denominados de parâmetros ou dimensões da qualidade. A quantidade de dimensões existentes varia de acordo com os autores, sendo que as dimensões mais exploradas na literatura são a acurácia, a completude, a consistência lógica e a confiabilidade [Firmani et al. 2016]. Este trabalho apresenta a ferramenta QualiOSM, com o objetivo de melhorar a dimensão da completude, representada como a proporção entre a presença de metadados associados a um conjunto de objetos em comparação com o total de objetos desse conjunto [Sehra et al. 2017].

Em grande parte dos SIGV, os usuários criam ou enviam conteúdo por meio da atribuição de etiquetas associadas aos objetos denominadas de *tags*. O processo de adição de etiquetas, também chamado de tagueamento ou marcação, foi descrito como um dos dilemas associados ao comportamento dos usuários na Web 2.0 e, dessa forma, existem vários estudos na literatura que exploraram esse mecanismo dentro de ferramentas colaborativas [Liu et al. 2011]. Por exemplo, [Codescu et al. 2011] organizaram uma ontologia com o objetivo de padronizar e facilitar a hierarquia de *tags* dentro da ferramenta de mapeamento colaborativo OpenStreetMap (OSM); [Mooney and Corcoran 2012] realizaram a análise de mais de 25.000 objetos na base de dados da Irlanda, Reino Unido, Alemanha e Áustria, identificando problemas no tagueamento de objetos do OSM; além disso, [Almendros Jiménez and Becerra Terón 2018] apresentaram um *framework* para a avaliação da qualidade dos dados também na ferramenta OSM, compreendendo um conjunto de métodos para analisar a qualidade do processo de atribuição de *tags*.

Diferentemente dos trabalhos citados anteriormente, este trabalho propõe a implementação da ferramenta QualiOSM com o objetivo de melhorar a qualidade das informações geográficas dentro da plataforma OpenStreetMap, sobretudo no que se refere ao processo de atribuição de *tags* de endereço aos objetos. Dessa forma, a intenção da ferramenta é contribuir com a completude das informações de endereço dos objetos dentro do OSM. A ferramenta foi testada em três diferentes cenários: ambiente urbano, ambiente rural e ambiente de favela.

O restante do artigo está estruturado da seguinte forma: A Seção 2 apresenta o desenvolvimento da ferramenta QualiOSM, com a descrição da metodologia e da arquitetura utilizadas para a realização do trabalho; a Seção 3 apresenta os resultados obtidos; por fim, a Seção 4 apresenta a conclusão e os trabalhos futuros.

2. QualiOSM

A ferramenta QualiOSM foi desenvolvida com o objetivo de melhorar a completude das informações de endereço associadas aos objetos da ferramenta OpenStreetMap. O aplicativo foi desenvolvido na forma de uma extensão (*plugin*) dentro do editor de dados JOSM (Java OpenStreetMap Editor)[2], responsável pelo maior número de edições em objetos dentro da plataforma do OpenStreetMap.

Para a implementação do adicionador de *tags* dentro da ferramenta QualiOSM, foi utilizada a técnica da geocodificação reversa, técnica em que a extração de informações textuais, como nome ou endereço, é realizada a partir de um par de coordenadas geográficas (latitude e longitude). Neste trabalho, foi utilizada a ferramenta Nominatim[3], a qual procura nomes e endereços nos dados do OSM a partir de um par de coordenadas geográficas, gerando os dados de endereço no formato XML (*Extensible Markup Language*) ou JSON (*JavaScript Object Notation*).

A Figura 1 apresenta a arquitetura utilizada para a implementação da ferramenta QualiOSM. Conforme pode ser observado, a arquitetura foi dividida em três camadas: a camada mais externa é a camada de apresentação, responsável por prover a interface entre o usuário e o editor de dados JOSM, além de fornecer o carregamento de imagens aéreas; o *plugin* QualiOSM juntamente com a funcionalidade do adicionador de *tags* foram desenvolvidos dentro da camada de aplicação, em que é possível observar também a interação com a API da ferramenta OpenStreetMap; por fim, a camada de dados é responsável por prover o gerenciamento dos dados do OpenStreetMap e fazer a interação com a ferramenta Nominatim. Dessa forma, quando um usuário seleciona um objeto ou conjunto de edifícios dentro do JOSM, a ferramenta Nominatim buscará as informações de endereço a partir do par de coordenadas e retornará essas informações para o adicionador automático de *tags* implementado dentro do *plugin* QualiOSM.

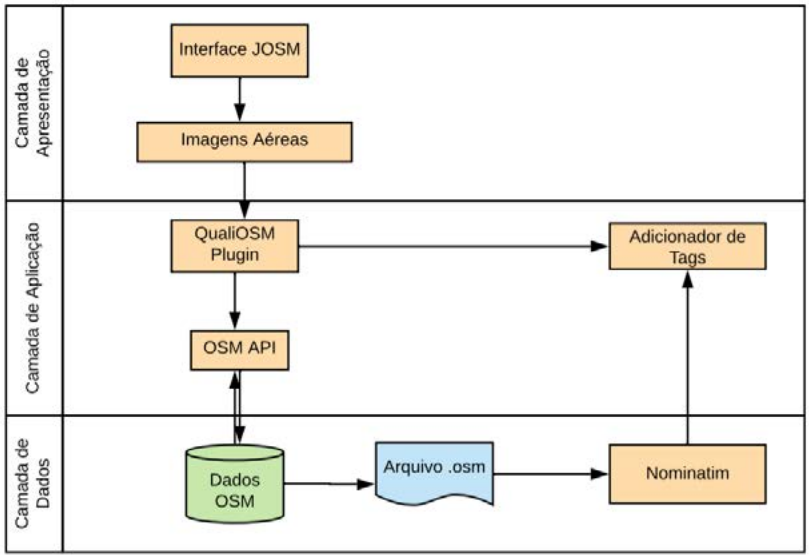


Figura 1. Arquitetura para implementação do aplicativo QualiOSM.

[2]https://josm.openstreetmap.de/ [Acesso em maio de 2020.]
[3]https://nominatim.openstreetmap.org/ [Acesso em maio de 2020.]

A partir da análise das *tags* de endereço mais utilizadas no OpenStreetMap observadas a partir das estatísticas presentes no site TagInfo[4] no mês de maio de 2020, foi realizada a escolha de incluir as quatro *tags* mais utilizadas dentro da ferramenta QualiOSM: *addr:street* (rua), *addr:city* (cidade), *addr:suburb* (bairro) e *addr:postcode* (código postal). Além disso, também foram incluídas na ferramenta a *tag addr:housenumber*, por ser a etiqueta de endereço mais utilizada dentro da ferramenta OpenStreetMap em geral, e a *tag addr:building*, por conter a informação com o nome de cada edifício. A partir dos dados do OpenStreetMap no Brasil, foram considerados três cenários distintos de teste para realizar a avaliação da ferramenta:

- Cenário I - ambiente urbano, considerando a região administrativa do Plano Piloto, na cidade de Brasília;
- Cenário II - ambiente rural, considerando a parte periférica das cidades interioranas de Mogi das Cruzes, Ribeirão Pires e Santo André, no estado de São Paulo;
- Cenário III - ambiente de favela, considerando a área pertencente à comunidade da Rocinha, no estado do Rio de Janeiro.

3. Resultados

Após a implementação do adicionador de *tags* dentro do editor de dados JOSM e a coleta de dados nos três cenários observados a partir de arquivos no formato *.osm* correspondentes às regiões de interesse, foram iniciados os testes da ferramenta. Dessa forma, o adicionador de *tags* foi acionado selecionando a predefinição "Contrução Humana/Edificação" dentro do editor JOSM e, em seguida, o adicionador foi aplicado nas três regiões de interesse. Feito isso, procedeu-se à análise das *tags* associadas aos objetos selecionados antes e após a atuação do adicionador de *tags*.

Aplicando o adicionador de *tags* em cenário urbano, foi escolhida a área da Região Administrativa do Plano Piloto, parte central da cidade de Brasília. Conforme pode ser observado a partir da Tabela 1, o resultado mostrou-se mais satisfatório em relação à inclusão da *tag* de código postal, uma vez que ocorreu um salto de 1,84% de edifícios associados para 97,14% de edifícios associados. Em relação à *tag addr:suburb* ocorreu um aumento de 1,76% para 41,83% de edifícios associados; Em relação à *tag addr:city* ocorreu um aumento de 2,12% para 45,46% de edifícios associados; Em relação à *tag addr:building*, ocorreu um aumento de 0% para 10,21% de edifícios associados. Não ocorreu nenhuma mudança em relação às *tags addr:street* (5,28% de edifícios associados) e *addr:housenumber* (1,59 % de edifícios associados) devido à falta de informações correspondentes na ferramenta Nominatim.

Aplicando o adicionador de *tags* em cenário rural, foi escolhida a área da parte periférica das cidades interioranas de Mogi das Cruzes, Ribeirão Pires e Santo André, no estado de São Paulo. Conforme pode ser observado a partir da Tabela 2, o resultado mostrou-se mais satisfatório em relação à inclusão da *tag* de código postal, em que houve um salto de 0% de edifícios associados para 100% de edifícios associados, e da *tag addr:city*, em que houve aumento de 3,25% para 100% de edifícios associados. Em relação à *tag addr:suburb*, houve um aumento de 0% para 79,86%. Não ocorreu mudança em relação às *tags addr:street* (3,25% de edifícios associados), *addr:housenumber* e *addr:building* (nenhum edifício associado).

[4]https://taginfo.openstreetmap.org/ [Acesso em maio de 2020.]

Tabela 1. Inclusão de *tags* de endereço em cenário urbano.

Tag	Antes	Depois
addr:building	0%	10,21%
addr:city	2,12%	45,46%
addr:postcode	1,84%	97,14%
addr:housenumber	1,59%	1,59%
addr:street	5,28%	5,28%
addr:suburb	1,76%	41,83%

Aplicando o adicionador de *tags* em cenário de favela, foi escolhida a área pertencente à comunidade da Rocinha, no estado do Rio de Janeiro. Conforme pode ser observado a partir da Tabela 3, o resultado mostrou-se mais satisfatório em relação à inclusão da *tag* de código postal, em que houve um aumento de 0,36% de edifícios associados para 100% de edifícios associados, da *tag addr:suburb*, em que houve aumento de 0,71% para 100% de edifícios associados. No caso desse cenário, foi verificado um grande submapeamento dos edifícios, uma vez que até o dia 10 de maio de 2020, havia apenas 281 edifícios mapeados na comunidade da Rocinha dentro do OpenStreetMap, enquanto no censo do IBGE de 2010 já constavam mais de 26 mil edifícios na comunidade[5]. Dessa forma, pode-se dizer que ainda há potencial para que Organizações Não Governamentais e demais pessoas interessadas em mapeamento colaborativo contribuam para a melhoria do mapa em regiões correspondentes a favelas.

Tabela 2. Inclusão de *tags* de endereço em cenário rural.

Tag	Antes	Depois
addr:building	0%	0%
addr:city	3,25%	100%
addr:postcode	0%	100%
addr:housenumber	0%	0%
addr:street	3,25%	3,25%
addr:suburb	0%	79,86%

Tabela 3. Inclusão de *tags* de endereço em cenário de favela.

Tag	Antes	Depois
addr:building	0%	3,2%
addr:city	0,71%	100%
addr:postcode	0,36%	100%
addr:housenumber	0,71%	0,71%
addr:street	0,71%	0,71%
addr:suburb	0,71%	100%

[5]http://www.rocinha.org/noticias/rocinha/view.asp?id=895 [Acesso em maio de 2020.]

4. Conclusão

A ferramenta do adicionador de *tags* demonstrou potencial para a melhoria da dimensão da completude para informações de objetos dentro da ferramenta colaborativa OpenStreetMap, porém ainda necessita de aperfeiçoamentos. Além disso, nos três cenários analisados, o adicionador de *tags* mostrou-se mais eficiente para a inclusão da *tag* de código postal, que é uma etiqueta importante para a localização de edifícios dentro do OpenStreetMap.

Conforme os resultados observados, percebe-se que ocorreu um significativo aumento em termos percentuais em relação a objetos associados às *tags addr:city* e *addr:suburb*, contribuindo para a completude desse tipo de informação dentro da ferramenta. Foi observado também que, em geral, regiões rurais e periféricas apresentam menos informações mapeadas comparando-se com regiões de grandes centros urbanos, confirmando a heterogeneidade de dados usualmente presente em ferramentas colaborativas.

Como trabalho futuro, pretende-se explorar outras *tags* além das *tags* de endereço abordadas neste trabalho, fazendo uso de outras ferramentas além da ferramenta Nominatim para a localização de informações. Também pretende-se testar a ferramenta em outros cenários e avaliar outras dimensões da qualidade em sistemas colaborativos, tais como a consistência lógica e a acurácia.

Referências

Almendros Jiménez, J. M. and Becerra Terón, A. (2018). Analyzing the tagging quality of the spanish OpenStreetMap. *ISPRS International Journal of Geo-Information*, 7(8):323.

Codescu, M., Horsinka, G., Kutz, O., Mossakowski, T., and Rau, R. (2011). Osmonto-an ontology of OpenStreetMap tags. *State of the map Europe (SOTM-EU)*, 2011.

Firmani, D., Mecella, M., Scannapieco, M., and Batini, C. (2016). On the meaningfulness of "Big Data quality". *Data Science and Engineering*, 1(1):6–20.

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221.

ISO, I. (2013). 19157: 2013: Geographic information—data quality. *International Organization for Standardization: Geneva, Switzerland*.

Liu, D., Wang, M., Hua, X.-S., and Zhang, H.-J. (2011). Semi-automatic tagging of photo albums via exemplar selection and tag inference. *IEEE Transactions on Multimedia*, 13:82–91.

Mooney, P. and Corcoran, P. (2012). The annotation process in OpenStreetMap. *Transactions in GIS*, 16(4).

Sehra, S. S., Singh, J., and Rai, H. S. (2017). Assessing OpenStreetMap data using intrinsic quality indicators: an extension to the QGIS processing toolbox. *Future Internet*, 9(2):15.

Tomlinson, R. F. (2007). *Thinking about GIS: geographic information system planning for managers*, volume 1. ESRI, Inc.

JSONGlue: A hybrid matcher for JSON schema matching

Vitor Marini Blaselbauer[1], **João Marcelo Borovina Josko**[1]

[1]Center of Mathematics, Computing and Cognition – Federal University of ABC (UFABC)
Av. dos Estados, 5001 – Santo Andre – SP – Brazil

vitor.blaselbauer@aluno.ufabc.edu.br, marcelo.josko@ufabc.edu.br

Abstract. *The JSON-based databases are the dominant cloud-oriented approach to handle structured-variable data. Its schemaless nature makes writing operations quick at the expense of integrating data with heterogeneous schemas. Schema matching literature collects various contributions to identify similarities in the XML documents, but very few for JSON. This work introduces a hybrid supervised matcher (named JSONGlue) that executes linguistic, semantic, and instance-based methods in parallel to match multiple heterogeneous JSON schemas. It also reports JSONGlue's first results and presents its next evolution steps.*

1. Introduction

The JSON-based databases are the dominant cloud-oriented approach to handle structured-variable data in Big Data or Internet contexts. Besides their scalable feature, their no rigid schema before writing into the database made application development more straightforward. Conversely, since the database ignores the structure of their persisted data, applications may store semantically equivalent documents using heterogeneous schemas. This heterogeneity increases the complexity of integrated data access by analytical procedures and makes database evolution hard.

Schema matching denotes the methods of identifying a possible correspondence between elements of different schemas that contain semantically related data [Gal 2011]. Its literature presents important methods for various data representations, including structured and XML. Several studies discussed approaches that focus on schema characteristics (e.g., attributes names) or instance-level similarity (e.g., data distribution) to determine the schemas correspondence [Gal 2011]. Other works adopt various methods (hybrid approach) to identify similarities, whereas some studies add the semantic distinction capacity of human beings into the matching process [Gal 2011]. However, few researchers have addressed schema matching for the JSON data format. Indeed, we have found two works [Padilha 2020, Waghray 2020] whose purpose or design differ from the ones used by the present work.

This work introduces a hybrid matcher (named JSONGlue) that uses linguistic, semantic, and data distribution approaches to match several JSON schemas of a given dataset. Our supervised matcher maps all matches and corresponding similarity measures to a graph structure to enable future human supervision and analysis mediated by visualizations.

This paper is structured as follows: In Section 2, we briefly discuss the main differences between JSON and XML. Next, we characterize the architecture and components of our matcher in Section 3 and report its preliminary results in Section 4. Finally, we review related works in Section 5 and present conclusions and future works in Section 6.

2. Characterizing JSON and XML

JSON (JavaScript Object Notation) and XML (Extensible Markup Language) are both popular notations to store and transfer data. Although they share several aspects, the specificity of JSON makes its matching process slightly different from XML.

JSON has an array data type that might include any of the JSON value types (e.g., string, numeric), embedded attributes, or even another array (nested array). Moreover, JSON allows the same element label to occur in the same schema as it does not have a namespaces feature. This situation may lead to label collision. Finally, the unordered and arbitrary levels of nesting key-value pairs of JSON documents contrast with the essential role of the order of XML elements.

3. JSONGlue Characteristics

Figure 1 exhibits the JSONGlue architecture style and its modules' communication flow. These modules code uses Python language due to its extensive support libraries that facilitate string operations, manipulation of massive amounts of data, parallel processing, among other possibilities. Except for the normalization module that builds the graph, the remaining modules run parallelly using all core available. For each distinct pair of JSON schemas, our matcher spawns a process to run the three schema matching modules.

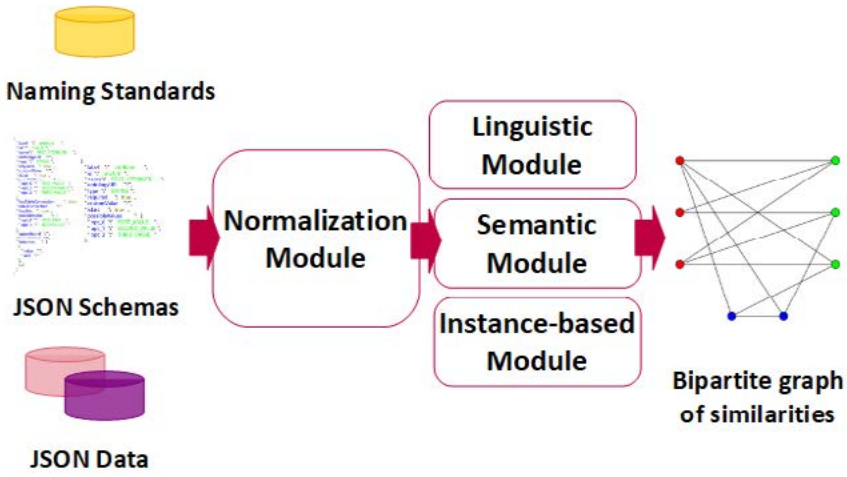


Figure 1. JSONGlue architecture style and components (Source: The authors)

The *normalization module* has three steps executed in sequence. The first is in charge of removing irrelevant characters, including numbers, extra spaces between tokens, special symbols, punctuation symbols, and stop words based on WordNet [Miller 1998]. In turn, the second transforms elements' text to lowercase and convert abbreviations to the corresponding business name. This last procedure occurs if there is a file that represents this organization's naming standard convention. The third step builds a disconnected graph $\mathcal{G}$ composed of subgraphs $\mathcal{S}_i$, $\mathcal{S}_i = \mathcal{S}_j$ iff $i = j$, that correspond to each input scheme $\mathcal{SC}_i$, $i \geq 2$. Moreover, this step breaks arrays of embedded attributes into individual elements for handling for the next steps.

The *linguistic module* is in charge of measuring the string similarity between all the JSON elements of all previous schemes loaded. In other words, this module applies a similarity function $ling : (a, b) \rightarrow [0, 1]$ (for each pair of nodes (a_k, b_m), $k = |\mathcal{S}_i|$,

$a_k \in \mathcal{S}_i$, $m = |\mathcal{S}_j|$, $b_m \in \mathcal{S}_j$, $i \neq j$. Afterward, it creates an edge for each pair (a_k, b_m) whose linguistic property contains this function's result. This work uses the Jaro-Winkler algorithm as it is best suited for comparing short strings [Peng et al. 2012], though it produces favorable ratings when strings beginnings are the same.

The *semantic module* uses a lexical function to measure the similarity between all pairs of nodes. This function uses the WordNet monolingual database because of its widespread adoption for linguistics processing tasks [Miller 1998]. The current JSONGlue version assumes no semantic available about data (e.g., a partial ontology or data dictionary). Hence, its lexical function $sema : (a, b) \rightarrow_{max} [0, 1]$ transverses all WordNet synset (set of synonyms denoting the same concept) until it reaches the one of maximum similarity for each pair of nodes (a_k, b_m). When comparing the JSON elements with a different number of words, this function reconciles their length using the ancestor (when available) or assume no similarity between head nouns. Subsequently, it returns the average between the individual similarity of each modifier and head noun. In the current version, our tool does not handle the JSON elements with three or more words. Analogous to the previous module, our system connects each pair of nodes and assign to its similarity property the lexical function's result. We use the Wu-Palmer measure in which similarity is inverse to the path distance between two concepts [Miller 1998].

Lastly, the instance-based module applies three functions to measure how similar data values are. The first $diffAVG : (a, b) \rightarrow R_{\geq 0}$ calculates the difference between each pair of nodes by considering the average length of their data values, while the second $diffSTD : (a, b) \rightarrow R_{\geq 0}$ does the same for the standard deviation. The third function $distHIST : (a, b) \rightarrow R_{\geq 0}$ measures the distance between two histograms H_a and H_b that describe the characters' frequency within the data values of a given pair of nodes. The calculation of this distance considers both the overlapping and non-overlapping parts of the histograms (D_1 algorithm by [Cha and Srihari 2002]).

4. Case Study with JsonGlue

4.1. Data settings

This case study used artificial and real data. For the former, our algorithm considered a reduced customer invoice business domain (outlined below) to generate three datasets of increasing sizes (Figure 4). Each dataset (D_1, D_2, D_3) has a distinct JSON schema (S_1, S_2, S_3, respectively). It randomly introduced schema variations in the number of attributes (15 on average) and their grouping (e.g., array, embedded), nomenclature (e.g., names with and without abbreviations), and position of attributes. Such an algorithm also randomly generated the values of the attributes with several differences (e.g., size, range of values).

Customer (CustomerID, Name, Address, Phone, Email)
Invoice (InvoiceID, CustomerID, InvoiceDate, DeliveryAddress, TotalAmount)
Invoice Item (InvoiceID, InvoideItemID, ProductName, ItemQuantity, ItemTotal)

For the latter, we extract two schemas (25 attributes on average) from a COVID-19 dataset[1] with the latest numbers from every US territory. Is it worth noting that we used the real dataset for validation purpose and the algorithm-generated datasets for validation and discussion of isolated methods results.

[1]The COVID Tracking Project

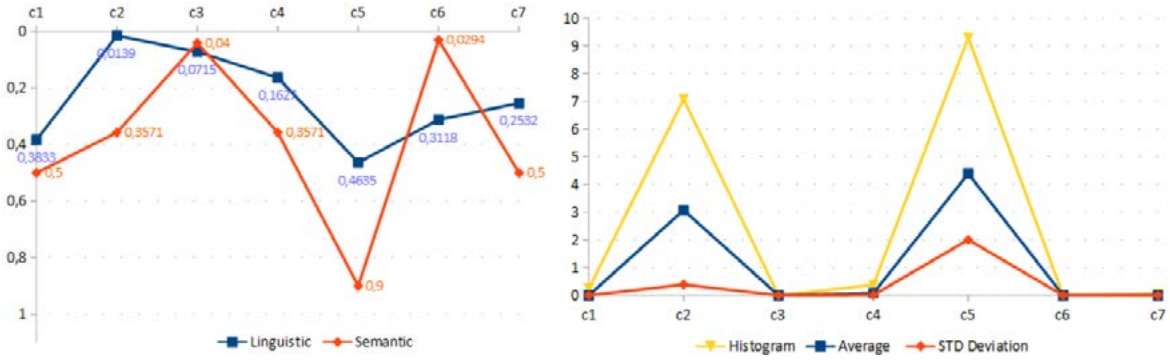
4.2. Methods Results and Validation

Table 1 shows some selected mapping cases of the JSONGlue outcome to highlight the characteristics of its methods. Each pair of lines represent a single case preceded by an identification code (c_n, $n \geq 1$) to make its reference easy. In turn, the columns present the schemas compared, its original elements names, and the names of the elements used on linguistic and semantic comparisons after normalization and ancestor identification. It is worth noting that linguistic and semantic measures (Figure 2a) use *zero* to denote full similarity and *one* otherwise.

Table 1. Selected Mapping Cases - Algorithm-based datasets

	Sch	Schema Element	Linguistic Element	Semantic Element
c_1 :	S_1 ·	[customer_phonenumber] ·	[customer phonenumber] ·	[customer phonenumber] ·
	S_2	[phone]	[phone]	[customer phone]
c_2 :	S_1 ·	[total_amount] ·	[total amount] ·	[total amount] ·
	S_2	[total_mount]	[total mount]	[total mount]
c_3 :	S_1 ·	[product_name] ·	[product name] ·	[product name] ·
	S_2	[product_title]	[product title]	[product title]
c_4 :	S_1 ·	[customer_name] ·	[customer name] ·	[customer name] ·
	S_3	[cus_name]	[cus name]	[cus name]
c_5 :	S_2 ·	[product_title] ·	[product title] ·	[product title] ·
	S_3	[inv_prod1]	[inv prod]	[inv prod]
c_6 :	S_1 ·	[invoice_id] ·	[invoice identifier] ·	[invoice identifier] ·
	S_2	[stmt_id]	[statement identifier]	[statement identifier]
c_7 :	S_1 ·	[quantity] ·	[quantity] ·	[invoice items quantity] ·
	S_2	[item_qty]	[item quantity]	[item quantity]

Semantic matching provides relevant results when schemas elements follow a naming standard or share correlated business terms (c_3, c_6 in Figure 2a). However, the outcome of this matching method is negatively affected by the absence of such concerns (c_4, c_5 in Figure 2a, respectively) or irregular nomenclature, including misspelling (c_2 in Figure 2a), excessive concatenation (c_1 in Figure 2a), or proper nouns.



(a) Linguistic and Semantic similarities **(b)** Instance-level distance

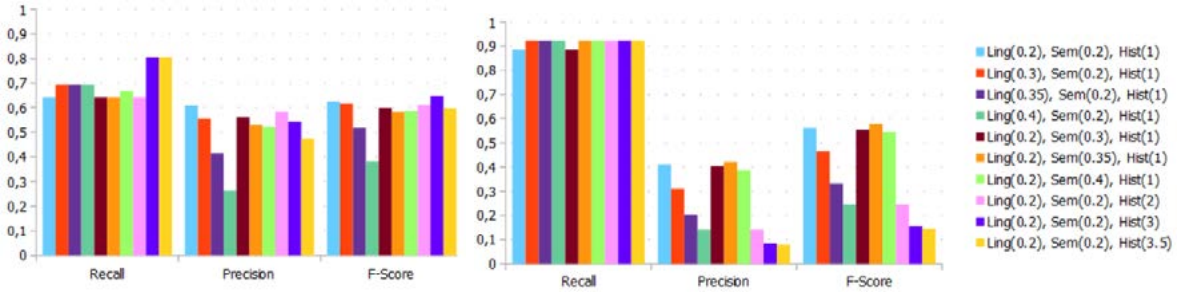
Figure 2. Similarity Results per Selected Case (Source: The authors)

Linguistic method outcomes can complement semantic similarity measurement to support matching analysis, although it also benefits from name homogeneity (c_3). For the

cases c_2 and c_4 (Figure 2a), the former provides a more robust indication of similarity than the latter. However, there are situations (c_6 in Figure 2a) that linguistic outcome is worse, or (c_1, c_7 in Figure 2a) both methods' measurements are uncertain for a proper decision.

Finally, instance-based methods can provide additional support for the previous analysis. In several cases, these methods reveal the little variance between the values of the elements compared, as observed in c_1, c_3, c_4, c_6, c_7 in Figure 2b. However, as the JSON elements may have a marked heterogeneity in terms of length, instance-based methods that use frequency of characters or string length distance decrease its certainty (c_2, c_5 in Figure 2b).

We validate JSONGlue matching results using a gold standard manually built for all datasets (Section 4.1). We also manually select some thresholds for each measure (the legend in Figure 3) as JSONGlue does not have an automatic threshold estimation yet.



(a) Algorithm-based dataset (Average) $60K$ **(b)** Covid Dataset

Figure 3. Matching Results per Set of Thresholds (Source: The authors)

Figure 3a reveals that the instanced-based method (histogram distance) tend to provide better results for datasets with high or moderate heterogeneity attribute nomenclature. The increase in its recall ($\simeq 25\%$) and f-score ($\simeq 5\%$) measures somehow followed its threshold growth. In contrast, Figure 3b illustrates that semantic methods tend to offer a superior results when schemas attributes align with business terms. Its f-score and precision measures are close as the threshold increases. Figure 3b also shows that the precision measure of the instance-based methods decreases dramatically for datasets whose attributes share very close data ranges.

The parallel matching support reduced 27% (on average) the comparison time for the datasets from $1k$ to $60k$. Such reduction increases a little more ($\simeq 42\%$ on average) for the datasets bigger than $60K$, as illustrated by Figure 4. The overall matching time reduction using parallel processing support was 35% on average. We used an Intel $i5-8500$ computer with six cores, 16GB of RAM, and Ubuntu 20.04 of 64 bits to gather all matching results.

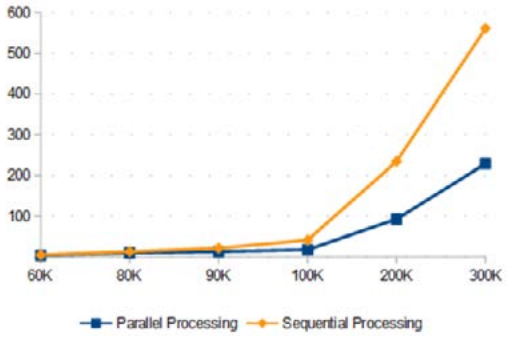


Figure 4. Matching time consumption (in min) per dataset size (Source: The authors)

5. Related Works

The literature regarding handling heterogeneous schema has a broad set of contributions that apply different perspectives to diverse data formats. Due to space restrictions, this work only discusses papers focusing on integrating heterogeneous JSON schemas.

Some studies apply one integration method to combine JSON documents on query time [Gallinucci et al. 2018] or to transform them into a relational representation [DiScala and Abadi 2016]. In another perspective, [Waghray 2020] shows that XML schema matching approaches do not readily support JSON schema matching.

The closest work to ours also combines linguistic, semantic, and instance-level methods to match heterogeneous schemas [Padilha 2020]. However, our work differs in the design approach and some of the methods adopted. For example, [Padilha 2020] does not use graphs to represent similarities among schemas, parallel matching, or histograms to compare data values. Conversely, our system does not calculate a summary similarity measure because we intend to integrate progressive matching visualization features.

6. Conclusion

This work reports the design approach and components of the JSONGlue system that matches JSON schemas using parallel processing. This matching process applies linguistic, semantic (single and compound noun), and instance-based methods to identify JSON elements' similarities represented in a graph form.

Nevertheless, JSONGlue neither consider semantic information nor uses human semantic distinction on the matching process. As future works, we intend to integrate human supervision and knowledge representations (e.g., expert rules, ontologies) to increase the JSONGlue matching performance.

References

Cha, S.-H. and Srihari, S. N. (2002). On measuring the distance between histograms. *Pattern Recognition*, 35(6):1355–1370.

DiScala, M. and Abadi, D. J. (2016). Automatic generation of normalized relational schemas from nested key-value data. In *Proceedings of the 2016 International Conference on Management of Data*, pages 295–310.

Gal, A. (2011). *Uncertain schema matching*. Morgan & Claypool Publishers, 1st edition.

Gallinucci, E., Golfarelli, M., and Rizzi, S. (2018). Variety-aware olap of document-oriented databases. In *DOLAP*.

Miller, G. A. (1998). *WordNet: An electronic lexical database*. MIT press.

Padilha, R. J. (2020). Um processo para casamento de esquemas de documentos json baseado na estrutura e nas instâncias. Master's thesis, Federal University of Santa Maria, Brazil.

Peng, T., Li, L., and Kennedy, J. (2012). A comparison of techniques for name matching. *GSTF Journal on Computing (JoC)*, 2(1):55–61.

Waghray, K. (2020). Json schema matching: Empirical observations. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2887–2889.

Modelagem Entidade-Relacionamento com TerraER

(*Distinguished Demo*)

Ricardo Terra

Departamento de Ciência da Computação
Universidade Federal de Lavras (UFLA)

terra@ufla.br

Abstract. *The Entity-Relationship (ER) model is widely used for teaching conceptual data modeling. However, existing tools usually do not reflect exactly what is taught in the classroom. TerraER is a free open-source learning tool designed to aid students in the creation of ER models. Our main goal is to provide students with a tool that reflects exactly the data modeling concepts learned in the classroom. In addition to supporting students in the creation of ER models, the tool also checks connections as soon as they are added to the model and—when invalid—notifies the student and also suggests the correct connections. As the main contribution, the tool seeks to make the learning process faster and more interactive for the student, besides reducing the teacher's correction effort.*

Resumo. *O modelo Entidade-Relacionamento (ER) é largamente adotado no ensino de modelagem de dados conceitual. No entanto, observou-se que as ferramentas existentes não refletem exatamente o que é ensinado em sala de aula. TerraER é uma ferramenta de aprendizagem de código aberto gratuita projetada para refletir exatamente os conceitos de modelagem de dados aprendidos em sala de aula. Além de apoiar os alunos na criação de modelos ER, a ferramenta também verifica conexões logo que são adicionadas ao modelo e – caso inválidas – não só notifica o aluno como sugere as conexões corretas. Como contribuição, busca-se tornar o processo de aprendizado mais rápido e interativo para o aluno, além de reduzir o esforço de correção pelo professor.*

É importante mencionar que este artigo condensa material das seguintes publicações:

> Cody Malnor; André Chateaubriand; Obede Carvalho; Ricardo Terra. Validação de Modelos ER. In *XXVI Workshop sobre Educação em Computação (WEI)*, páginas 1-10, 2018.
>
> Henrique Rocha; Ricardo Terra. TerraER – an Academic Tool for ER Modeling. *Methods and Tools*, 1(3):38-41, 2013.
>
> Henrique Rocha; Ricardo Terra. TerraER: Uma Ferramenta voltada ao Ensino do Modelo de Entidade-Relacionamento. In *VI Escola Regional de Banco de Dados (ERBD)*, páginas 1-4, 2010.

1. Introdução

A modelagem de dados é o principal componente do projeto conceitual do banco de dados. Dentre as técnicas existentes para essa modelagem, a técnica entidade-relacionamento (ER) – apresentada em 1976 por Peter Chen [3] – é ainda largamente utilizada principalmente pela sua simplicidade e legibilidade, produzindo um modelo que seja inteligível tanto pelo projetista do banco de dados quanto pelo usuário final [9, 5, 1, 2].

Em razão disso, grande parte das instituições de ensino superior utilizam o modelo ER no ensino de modelagem de dados conceitual. No entanto, nota-se uma carência em relação a ferramentas que utilizem a notação de Chen estendida e que tenham foco no modelo conceitual. Em razão disso, professores vêm adotando ferramentas voltadas para o modelo lógico como `DBDesigner`[1], `ERWin`[2], `Dia`[3], `EERCASE`[4] e `MySQLWorkbench`[5].

A adoção dessas ferramentas, mesmo estáveis e populares, não favorece ao aluno, uma vez que o aluno pratica o que lhe foi ensinado em uma ferramenta voltada a um outro modelo e que não possui fins acadêmicos. A partir dessa motivação, foi desenvolvida a ferramenta `TerraER` com o intuito de cobrir essa carência acadêmica. O objetivo principal da ferramenta é prover aos professores uma ferramenta mais voltada ao conteúdo lecionado e prover aos alunos uma ferramenta que estimule o seu aprendizado. Além de apoiar os alunos na criação de modelos ER, a ferramenta também verifica conexões logo que são adicionadas ao modelo e – caso inválidas – não só notifica o aluno como sugere as conexões corretas.

O restante deste artigo está organizado conforme descrito a seguir. A Seção 2 apresenta a ferramenta `TerraER`. A Seção 3 descreve a funcionalidade de validação de modelos. A Seção 4 enumera as contribuições sob a perspectiva educacional. E, por fim, a Seção 5 apresenta as considerações finais e os trabalhos futuros.

2. TerraER

O `TerraER` é uma ferramenta de código aberto – utilizada em mais de 30 instituições de ensino superior desde 2009 – voltada ao aprendizado de disciplinas de modelagem conceitual de banco de dados [7, 8, 6]. `TerraER` permite criar modelos conceituais de alto nível mais condizentes ao que os professores lecionam na disciplina de banco de dados. Isso pode ser constatado através da Figura 1, na qual a barra de ferramentas de objetos possui atalhos para criação de elementos do diagrama ER na notação de Peter Chen e EER, adotada por Elmasri e Navathe [5].

Instalação: O TerraER é distribuído em um único arquivo JAR disponível publicamente para *download*.[6] O arquivo JAR é autocontido, ou seja, pode ser colocado em qualquer pasta, não requer a instalação de bibliotecas adicionais e não altera os arquivos do sistema operacional (por exemplo, registro do Windows). Resumindo, TerraER requer apenas um *Java Runtime Environment* (JRE) previamente instalado no computador de destino.

Principais funcionalidades: Em seu último *release* – TerraER 3.13 – destacam-se as seguintes funcionalidades:

- criação de Modelos ER;
- validação de Modelos ER com sugestões de potenciais correções;
- persistência em arquivos XML;

[1] http://www.fabforce.net/dbdesigner4
[2] https://erwin.com/products/erwin-data-modeler/
[3] http://www.dia-installer.de
[4] https://www.sites.google.com/a/cin.ufpe.br/eercase/apresentacao
[5] https://www.mysql.com/products/workbench/
[6] http://www.terraer.com.br/

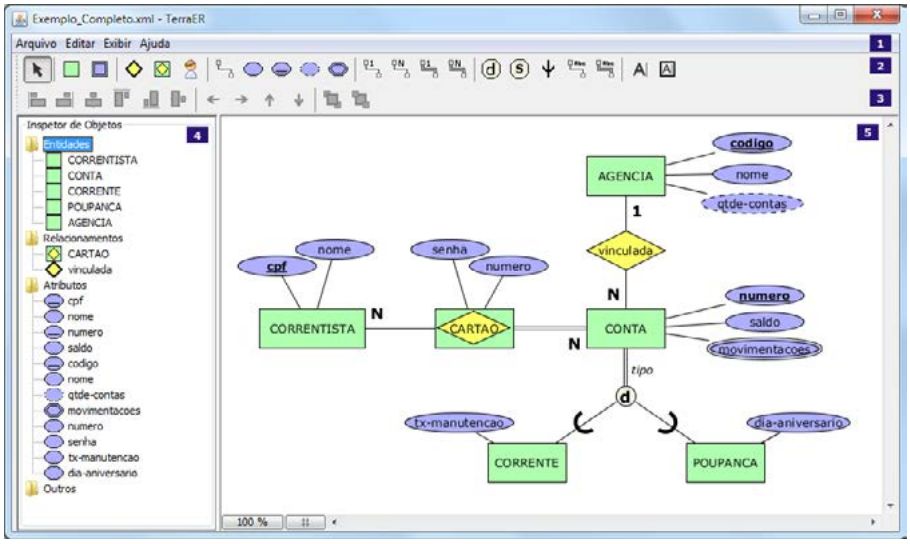


Figura 1. Modelo ER estendido na ferramenta TerraER

- exportação para PNG e impressão de modelos;
- funcionamento nos sistemas operacionais mais populares (multiplataforma);
- recurso desfazer/refazer; e
- modo de edição rápido.

É importante ressaltar que se trata de um projeto de código aberto, portanto encoraja-se os usuários a contribuir diretamente para o projeto TerraER[7]). Por exemplo, um aluno de graduação contribuiu para o projeto desenvolvendo a internacionalização para a língua inglesa.

Interface do usuário e descrição do recurso: A interface gráfica do usuário foi desenvolvida para ser prática, inteligível e intuitiva (ou seja, fácil de aprender e usar). Como ilustrado na Figura 1, a interface da ferramenta é dividida em cinco principais áreas:

1. *Barra de menu*: fornece aos usuários opções básicas, como abrir, salvar e imprimir os modelos. Mais importante, os modelos são salvos em formato XML, o que contribui diretamente para o recurso de multiplataforma. Na prática, os modelos salvos em um sistema operacional podem ser carregados em um sistema diferente sem problema algum;
2. *Barra de ferramentas de objetos*: fornece comandos para criar elementos do modelo ER – como as figuras e conexões ilustradas na Tabela 1 – na notação de Chen, conforme adotado por Elmasri e Navathe [5];

Tabela 1. Elementos ER

Figuras		*Conexões*	
Entidade	Entidade Fraca	Conexão de Atributo	Opcional ‘1 para’
Entidade Relacionamento	Relacionamento	Obrigatória ‘1 para’	Opcional ‘n para’
Relacionamento Fraco	Atributo	Obrigatória ‘n para’	Geral opcional
Atributo Chave	Atributo Chave Parcial	Geral obrigatória ‘n para’	Generalização
Atributo Derivado	Atributo Multivalorado		
Disjunção	Sobreposição		
União			

[7]https://github.com/rterrabh/TerraER

3. *Barra de ferramentas de posição*: fornece aos usuários meios para manipular a posição dos elementos – como alinhamento, sobreposição, etc. – para permitir uma formatação elegante dos modelos criados;
4. *Inspetor de objetos*: lista os elementos do modelo ER atual e permite que o usuário os selecione, remova ou edite. Na prática, esse recurso fornece uma maneira rápida e precisa de lidar com os elementos do modelo. Como um outro exemplo de contribuição, o desenvolvimento desse inspetor de objetos foi motivado pela sugestão de um aluno que teve dificuldade em localizar objetos específicos; e
5. *Área de desenho*: mostra a visão gráfica do modelo ER em criação. O usuário pode adicionar e remover elementos do modelo. Existe um recurso de *zoom*, que pode ser muito útil ao lidar com modelos grandes. Além disso, há um recurso de grade que auxilia os usuários a posicionar os elementos.

3. Validação de Modelos ER

Claramente, alunos não podem realizar qualquer tipo de conexão entre quaisquer dois elementos. Em um trabalho anterior [6], após uma formalização de cada elemento ER e as suas construções válidas, foi incorporado no `TerraER` um módulo de validação de modelos. A Figura 2 ilustra um grafo onde os vértices são figuras ou grupo de figuras e as arestas são rotuladas com as conexões válidas entre elas.

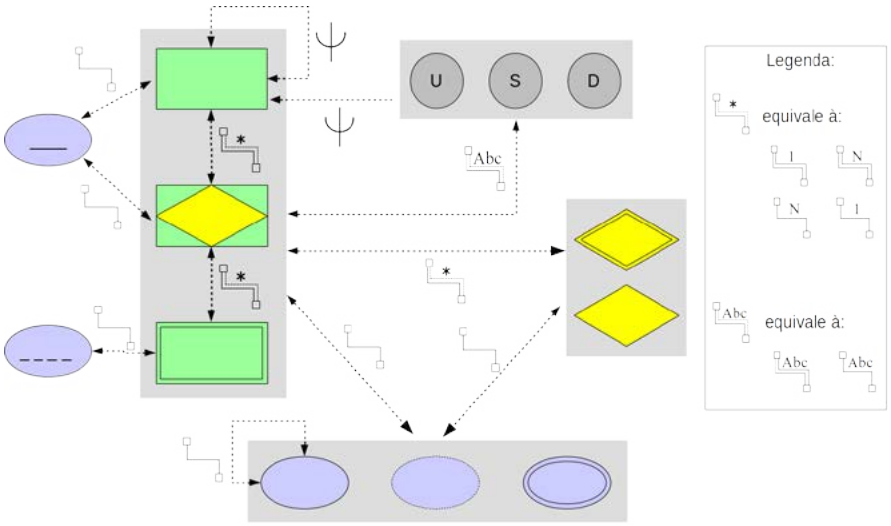


Figura 2. Grafo de conexões válidas em modelos ER [6]

Por exemplo, a conexão de atributo pode ser estabelecida entre (i) qualquer elemento do grupo composto por entidade, entidade relacionamento e entidade fraca e (ii) qualquer elemento do grupo composto por atributo, atributo derivado e atributo multivalorado.

Caso a conexão não seja permitida entre tais figuras, `TerraER` destacará a conexão na cor vermelha no intuito de alertar o aluno sobre o erro encontrado. Mais importante, o aluno não só é notificado do problema como pode solicitar sugestão de como corrigi-lo. Uma tela de recomendações, como a ilustrada na Figura 3, aponta quais as possíveis conexões entre as tais duas figuras e também quais as possíveis figuras que podem ser conectadas usando tal conexão.

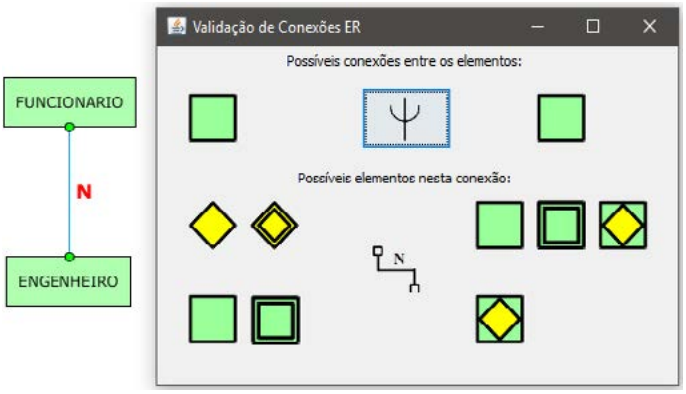


Figura 3. Exemplo de validação de um modelo ER

Por um lado, é exibida uma lista de conexões válidas entre os dois elementos envolvidos no erro encontrado. Por exemplo, conforme ilustrado na Figura 3, uma entidade □ está sendo erroneamente conectada à uma outra entidade □ por meio de uma conexão de relação N. Nesse caso, é exibido ao aluno que a única conexão válida entre duas entidades é a conexão de generalização ⊥.

Por outro lado, é também exibida uma lista de elementos que podem ser conectados usando a conexão em que há o erro (conexão de relação N, nesse exemplo). Por exemplo, conforme ilustrado na Figura 3, a conexão de relação pode ocorrer (i) relacionamento ◇ ou relacionamento fraco ◈ com entidade □, entidade fraca ▣ ou entidade relacionamento ⊠, e (ii) entidade □ ou entidade fraca ▣ com entidade relacionamento ⊠.

4. Contribuições sob a Perspectiva Educacional

Esta seção enumera as principais três contribuições relevantes do uso do `TerraER` sob a perspectiva educacional:

1. Sem gap *entre a ferramenta e a sala de aula:* O aluno pratica exatamente os conceitos aprendidos em sala de aula na ferramenta.

2. Permite o aluno errar e fornece feedback: A solução proposta envolve, no momento de cada nova conexão, uma verificação. Caso haja algum erro de construção, o aluno é notificado através de um *feedback* corretivo. Essa notificação pode ser considerada um *feedback* imediato por trazer à ciência do aluno o erro no momento em que é cometido, o que pode aprimorar ainda mais o aprendizado [4]. Ao permitir o erro do aluno e, em seguida, informá-lo de tal inconsistência, instiga-se o aprendizado a fim de evitar tal erro em atividades futuras.

3. Menor preocupação com detalhes do modelo: Um dos objetivos do módulo de validação é otimizar o processo educacional. Notificar e corrigir esses erros de construção facilitam a construção do modelo como um todo, uma vez que permite que o aluno foque na tarefa de modelagem conceitual, sem preocupações com formalizações do modelo. Isso ocorre sem sacrificar o aprendizado das formalizações do modelo, uma vez que o sistema de *feedback* provê exatamente isso. Ainda, essa facilitação também é transferida ao professor que poderá corrigir atividades focado exclusivamente na modelagem conceitual. Enfim, o fato de não ter que verificar o modelo fomenta uma maior eficiência no processo de educação.

5. Considerações Finais

Devido à sua simplicidade em representar dados e relações, o modelo Entidade-Relacionamento (ER) é largamente adotado no ensino de modelagem de dados conceitual. No entanto, observou-se que as ferramentas existentes não refletem exatamente o que é ensinado em sala de aula. Em razão disso, professores vêm adotando ferramentas voltadas para o modelo lógico que, mesmo sendo estáveis e populares, não favorecem a aprendizagem, uma vez que o aluno pratica o que lhe foi ensinado em uma ferramenta voltada a um outro modelo e que não possui fins didáticos.

Diante disso, este artigo apresenta TerraER que é uma ferramenta de aprendizagem de código aberto gratuita projetada para refletir exatamente os conceitos de modelagem de dados aprendidos em sala de aula. Além de apoiar os alunos na criação de modelos ER, a ferramenta também verifica conexões logo que são adicionadas ao modelo e – caso inválidas – não só notifica o aluno como sugere as conexões corretas. Como contribuição, busca-se tornar o processo de aprendizado mais condizente com a sala de aula, rápido e interativo para o aluno, além de reduzir o esforço de correção pelo professor.

O código do TerraER – junto com os últimos *releases* e um vídeo demonstrando a ferramenta – estão publicamente disponíveis em:

https://github.com/rterrabh/TerraER/

Agradecimentos: Este trabalho é apoiado pela FAPEMIG (APQ-03513-18) e CNPq (305829/2018-1).

Referências

[1] S. A. Korth H. F. and S. Sudarshan. *Sistema de Banco de Dados*. Elsevier, 5th edition, 2006.

[2] S. Bagui and R. Earp. *Database Design Using Entity-Relationship Diagrams*. CRC Press LLC, 1964.

[3] P. P. Chen. The entity-relationship model – towards a unified view of data. *ACM Transactions on Database Systems*, pages 9–36, Março 1976.

[4] R. Ellis. Corrective feedback and teacher development. *L2 Journal*, 1(1):3–18, 2009.

[5] R. Elmasri and S. B. Navathe. *Sistemas de Banco de Dados*. Pearson Addison Wesley, 6th edition, 2011.

[6] C. Malnor, A. Chateaubriand, O. Carvalho, and R. Terra. Validação de modelos ER. In *XXVI Workshop sobre Educação em Computação (WEI)*, pages 1–10, 2018.

[7] H. Rocha and R. Terra. TerraER: Uma ferramenta voltada ao ensino do modelo de entidade-relacionamento. In *VI Escola Regional de Banco de Dados (ERBD)*, pages 1–4, 2010.

[8] H. Rocha and R. Terra. TerraER - an academic tool for ER modeling. *Methods and Tools*, 1(3):38–41, 2013.

[9] T. Teorey, S. Lightstone, and T. Nadeau. *Projeto e Modelagem de Banco de Dados*. Elsevier, 2007.

Tutoriais
Tutorials

Blockchain System Foundations

Mohammad Javad Amiri Sujaya Maiyya Victor Zakhary
Divyakant Agrawal Amr El Abbadi
Department of Computer Science, University of California, Santa Barbara
Santa Barbara, California
{amiri,sujaya-maiyya,victorzakhary,agrawal,amr}@cs.ucsb.edu

1 INTRODUCTION

Bitcoin [21] is considered the first successful global scale peer-to-peer cryptocurrency. The Bitcoin protocol explained by the *mysterious Nakamoto* allows financial transactions to be transacted among participants without the need for a trusted third party, e.g., bank, credit card company, or PayPal. Bitcoin eliminates the need for such a trusted third party by replacing it with a distributed ledger that is fully replicated among all participants in the cryptocurrency system. This distributed ledger is referred to as *blockchain*.

Blockchain is a secure linked list of blocks containing financial transactions that occur in the system and linked by hash pointers. The main challenge that Bitcoin addresses is to maintain a consistent view of this replicated blockchain in a secure and fault-tolerant manner in a *permissionless* setting and in the presence of malicious participants. Unlike *permissioned* settings where all the participants in the system are known *a priori*, a permissionless setting allows participants to freely join and leave the system without maintaining any global knowledge of the number of participants. To address these challenges, Bitcoin builds on foundations developed over the last few decades from diverse fields [22], but primarily from the fields of **cryptography** [8, 24], **distributed systems** [10, 16, 17] and **data management** [9, 19, 27].

Bitcoin uses a notion of *miners* who need to perform a computationally challenging *Proof of Work (PoW)* puzzle before they can add any block of transactions to the replicated blockchain. Since the PoW puzzle is computationally hard, very few miners can successfully solve the puzzle, and hence a successful miner can add a block to the blockchain and be guaranteed, with very high probability, to be unique. Many concerns have been raised about the wasted massive energy requirements to *mine* one Bitcoin block. This mining approach to determine the process eligible to add a new block to the block chain is in contrast to the distributed systems approach, that has been promoting the use of Byzantine Agreement or consensus, which is efficient and more egalitarian. In fact, consensus protocols such as Paxos have been quite successful in recent years in laying the foundations of large global scale data management system. Unfortunately, Paxos has many limitations, especially from a global cryptocurrency point of view, including the requirement of a permissioned setting, and that participants can only fail by crashing. An alternative to Paxos that tolerates malicious failures is Practical Byzantine Fault-Tolerance (PBFT) [10]. Although it tolerates malicious failures, PBFT still requires a permissioned setting, and requires a large number of message exchanges, hence does not scale to the large number of participants expected in permissionless cryptocurrencies.

In this tutorial, our goal is to present to the database community an in-depth understanding of state-of-the-art solutions for efficient scalable blockchains. We progress towards this goal by starting from a detailed description of the protocols and techniques underlying the design of Bitcoin. Since most recent innovations in blockchain design depend critically on consensus protocols in malicious settings, we outline the basic foundations of distributed fault-tolerant consensus protocols. This is followed by a discussion of recent state-of-the-art permissioned blockchains. Since the participants are known and identified, permissioned blockchains can benefit from many techniques developed in the area of distributed computing over decades for reaching consensus, replicating state, and broadcasting transactions. We discuss various aspects of permissioned blockchains in the context of confidentiality [2, 7], verifiability [6, 20], performance [4, 7, 12, 15, 25, 26], and scalability [3, 5, 11, 13].

The wide adoption of permissionless open blockchain networks by both industry and academia suggests the importance of developing protocols and infrastructures that support peer-to-peer atomic cross-chain transactions. A two-party atomic cross-chain swap protocol was originally proposed by Nolen [1, 23] and generalized by Herlihy [14] to process multi-party atomic cross-chain swaps. Both Nolan's protocol and its generalization by Herlihy use smart contracts, hashlocks and timelocks to achieve atomic cross-chain swaps. These protocols require synchronous network assumptions and are not fault-tolerant. We therefore present a recently proposed atomic fault-tolerant cross chain protocol [28]. Finally, we give an overview of Fides [18], a database system that can detect malicious behaviour using blockchain.

2 ACKNOWLEDGEMENT

Partially funded by NSF grants CNS-1703560 and CNS-1815733.

REFERENCES

[1] 2018. Atomic cross-chain trading. https://en.bitcoin.it/wiki/Atomic_cross-chain_trading.

[2] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. CAPER: a cross-application permissioned blockchain. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1385–1398.

[3] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. On Sharding Permissioned Blockchains. In *Second International Conference on Blockchain*. IEEE, 282–285.

[4] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. ParBlockchain: Leveraging Transaction Parallelism in Permissioned Blockchain Systems. In *39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1337–1347.

[5] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. SharPer: Sharding Permissioned Blockchains Over Network Clusters. *arXiv preprint arXiv:1910.00765* (2019).

[6] Mohammad Javad Amiri, Joris Duguépéroux, Tristan Allard, Divyakant Agrawal, and Amr El Abbadi. 2020. SEPAR: A Privacy-Preserving Blockchain-based System for Regulating Multi-Platform Crowdworking Environments. *arXiv preprint arXiv:2005.01038* (2020).

[7] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. 2018. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *arXiv preprint arXiv:1801.10228* (2018).

[8] Adam Back. 2002. Hashcash-a denial of service counter-measure. (2002).

[9] Philip A Bernstein, Vassos Hadzilacos, and Nathan Goodman. 1987. Concurrency control and recovery in database systems. (1987).

[10] Miguel Castro, Barbara Liskov, et al. 1999. Practical Byzantine fault tolerance. In *OSDI*, Vol. 99. 173–186.

[11] Hung Dang, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi. 2019. Towards Scaling Blockchain Systems via Sharding. In *SIGMOD Int. Conf. on Management of Data*. ACM.

[12] Christian Gorenflo, Stephen Lee, Lukasz Golab, and Srinivasan Keshav. 2019. Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second. In *Int. Conf. on Blockchain and Cryptocurrency (ICBC)*. IEEE, 455–463.

[13] Suyash Gupta, Sajjad Rahnama, Jelle Hellings, and Mohammad Sadoghi. 2020. ResilientDB: Global Scale Resilient Blockchain Fabric. *Proceedings of the VLDB Endowment* 13, 6 (2020), 868–883.

[14] Maurice Herlihy. 2018. Atomic cross-chain swaps. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*. ACM, 245–254.

[15] Jae Kwon. 2014. Tendermint: Consensus without mining.

[16] Leslie Lamport et al. 2001. Paxos made simple. *ACM Sigact News* 32, 4 (2001), 18–25.

[17] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4, 3 (1982), 382–401.

[18] Sujaya Maiyya, Danny Hyun Bum Cho, Divyakant Agrawal, and Amr El Abbadi. 2020. Fides: Managing Data on Untrusted Infrastructure. In *40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE.

[19] C Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, and Peter Schwarz. 1992. ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. *ACM Transactions on Database Systems (TODS)* 17, 1 (1992), 94–162.

[20] JP Morgan. 2016. Quorum whitepaper. *En línia]. Available: https://github. com/jpmorganchase/quorumdocs/blob/master/Quorum% 20Whitepaper% 20v0* 1 (2016).

[21] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[22] Arvind Narayanan and Jeremy Clark. 2017. Bitcoin's academic pedigree. *Commun. ACM* 60, 12 (2017), 36–45.

[23] Tier Nolan. 2013. Alt chains and atomic transfers. https://bitcointalk.org/index.php?topic=193281.msg2224949/msg2224949.

[24] Ronald L Rivest, Adi Shamir, and Leonard Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* 21, 2 (1978), 120–126.

[25] Pingcheng Ruan, Dumitrel Loghin, Quang-Trung Ta, Meihui Zhang, Gang Chen, and Beng Chin Ooi. 2020. A Transactional Perspective on Execute-order-validate Blockchains. In *SIGMOD International Conference on Management of Data*. ACM, 543–557.

[26] Ankur Sharma, Felix Martin Schuhknecht, Divya Agrawal, and Jens Dittrich. 2019. Blurring the lines between blockchains and database systems: the case of hyperledger fabric. In *SIGMOD International Conference on Management of Data*. ACM, 105–122.

[27] Gerhard Weikum and Gottfried Vossen. 2001. *Transactional information systems: theory, algorithms, and the practice of concurrency control and recovery*. Elsevier.

[28] Victor Zakhary, Divyakant Agrawal, and Amr El Abbadi. 2020. Atomic commitment across blockchains. *Proceedings of the VLDB Endowment* 13, 9 (2020), 1319–1331.

Principles of Distributed Database Systems: spotlight on NewSQL

Patrick Valduriez
Inria, University of Montpellier, CNRS, LIRMM, France
LeanXcale, Spain

The first edition of the book Principles of Distributed Database Systems, co-authored with Prof. Tamer Özsu (University of Waterloo) appeared in 1991 when the technology was new and there were not too many products. In the Preface to the first edition, we had quoted Michael Stonebraker who claimed in 1988 that in the following 10 years, centralized DBMSs would be an "antique curiosity" and most organizations would move towards distributed DBMSs. That prediction has certainly proved to be correct, and most systems in use today are either distributed or parallel.

The fourth edition of this classic textbook [Özsu & Valduriez 2020] provides major updates, in particular, new chapters on big data platforms, NoSQL, NewSQL and polystores. In this tutorial, we introduce these major updates, with a focus on NewSQL.

NewSQL is the latest technology in the big data management landscape, enjoying a fast-growing rate in the DBMS and BI markets. NewSQL combines the scalability and availability of NoSQL with the consistency and usability of SQL. By providing online analytics over operational data, NewSQL opens up new opportunities in many application domains where real-time decision is critical. Important use cases are eAdvertisement (such as Google Adwords), IoT, performance monitoring, proximity marketing, risk monitoring, real-time pricing, real-time fraud detection, etc. NewSQL may also simplify data management, by removing the traditional separation between NoSQL and SQL (ingest data fast, query it with SQL), as well as between operational database and data warehouse / data lake (no more ETLs!). However, a hard problem is scaling out transactions in mixed operational and analytical (HTAP) workloads over big data, possibly coming from different data stores (HDFS, SQL, NoSQL). Today, only a few NewSQL systems have solved this problem, e.g., the LeanXcale NewSQL DBMS that has a highly scalable transaction management [Jimenez-Peris 2011] and a polystore [Kolev 2016a, Kolev 2016b].

A first in-depth presentation of NewSQL was given in a tutorial at IEEE Big Data 2019 with Prof. Ricardo Jimenez-Peris (CEO and founder at LeanXcale) [Valduriez 2019]. In this tutorial, we provide a taxonomy of NewSQL systems based on major dimensions including targeted workloads, capabilities and implementation techniques. We illustrate with popular NewSQL systems such as Google Spanner [Corbett 2012], LeanXcale [Kolev 2018], CockroachDB [Taft 2020], SAP HANA [Färber 2011], MemSQL [Chen 2016] and Splice Machine [https://splicemachine.com]. In particular, we give a spotlight on some of the more advanced systems. We also compare with major NoSQL and SQL systems, and discuss integration within big data ecosystems and corporate information systems, using polystores. Finally, we discuss the current trends and research directions.

References

[Chen 2016] J. Chen et al. The MemSQL Query Optimizer: A modern optimizer for real-time analytics in a distributed database. Proceeding of the VLDB 5PVLDB), 2016.

[Corbett 2012] J. C. Corbett. Spanner: Google's Globally-Distributed Database. USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2012.

[Färber 2011] F. Färber et al. SAP HANA database: Data management for modern business applications. ACM SIGMOD Record 40(4):45-51, 2011.

[Kolev 2016a] B. Kolev, P. Valduriez, C. Bondiombouy, R. Jiménez-Peris, R. Pau, J. Pereira. CloudMdsQL: Querying Heterogeneous Cloud Data Stores with a Common Language. Distributed and Parallel Databases, 34(4): 463-503, 2016.

[Kolev 2016b] B. Kolev, C. Bondiombouy, P.Valduriez, R. Jiménez-Peris, R. Pau, J. Pereira. The CloudMdsQL Multistore System. ACM SIGMOD Conference, 2016.

[Kolev 2018] B. Kolev, O. Levchenko, E. Pacitti, P. Valduriez, R. Vilaça, R. Gonçalves, R. Jiménez-Peris, P. Kranas. Parallel Polyglot Query Processing on Heterogeneous Cloud Data Stores with LeanXcale. IEEE BigData Conference, 2018.

[Jimenez-Peris 2011] R. Jimenez-Peris, M. Patiño-Martinez. System and method for highly scalable decentralized and low contention transactional processing. Filed at USPTO: 2011. European Patent #EP2780832, US Patent #US9,760,597.

[Özsu & Valduriez 2020] T. Özsu, P. Valduriez. Principles of Distributed Database Systems, 4th Edition, Springer, 2020.

[Taft 2020] R. Taft et al. CockroachDB: The Resilient Geo-Distributed SQL Database. ACM SIGMOD Conference, 2020.

[Valduriez 2019] P. Valduriez, R. Jimenez-Peris. NewSQL: principles, systems and current trends. IEEE Big Data Conference, 2019.

Palavras, apenas[1]: Métodos e Técnicas para Interfaces de Linguagem Natural em Bancos de Dados

Altigran da Silva, Paulo Martins, Brandell Ferreira e Lucas Citolin

Resumo

Desde os primórdios da tecnologia de Bancos de Dados nos anos 70, Interfaces de Linguagem Natural para Bancos de Dados (ILNBDs) têm sido uma aspiração quase utópica, tanto na academia quanto na indústria. De fato, apesar da larga adoção e popularidade dos bancos de dados nas últimas décadas, somente recentemente têm surgido na literatura métodos eficazes para o desenvolvimento de ILNBDs, que permitem que usuários sem conhecimento técnico possam explorar de forma efetiva os dados mantidos por Sistemas Gerenciadores de Bancos (SGBDs). Esse interesse renovado nas ILNBDs deve-se principalmente ao atual estágio de maturidade técnica de áreas como Aprendizagem de Máquina, Processamento de Linguagem Natural e Recuperação de Informação, cujos recententes avanços permitem a extração da semântica de palavras e sentenças textuais escritas por usuários com grande precisão e eficiência. Duas principais abordagens têm sido estudadas neste sentido. A primeira é a utilização de buscas por palavra-chave, de maneira similar ao que acontece em máquinas de busca. A segunda é o uso de sentenças escritas em linguagem natural para expressar consultas. Enquanto os sistemas baseados em palavra-chave oferecem uma forma mais simples e intuitiva de expressar consultas, os sistemas baseados em linguagem natural permitem expressar consultas mais complexas, envolvendo, por exemplo, agregações. Neste tutorial, apresentaremos uma visão geral de métodos e técnicas recentes que melhoraram em vários sentidos os algoritmos e modelos utilizados para construção de ILNBDs. O tutorial inclui ainda uma sessão de desenvolvimento, com exemplos práticos de código e bibliotecas usadas na construção de ILNBDs.

Referências

K. Affolter, K. Stockinger, A. Bernstein: A comparative survey of recent natural language interfaces for databases. VLDB J. 28(5): 793-819 (2019)

A. Afonso, A. da Silva, A., T. Conte, P. Martins, J. Cavalcanti, J., A. Garcia, LESSQL: Dealing with Database Schema Changes in Continuous Deployment. In 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 138-148).

C. Baik, H. V. Jagadish, and Y. Li. Bridging the semantic gap with SQL query logs in natural language interfaces to databases. In Proceedings of the 2019 IEEE International Conference on Data Engineering, pages 374–385, 2019.

F. Basik, B. Hättasch, A. Ilkhechi, A. Usta, S. Ramaswamy, P. Utama, N. Weir, C. Binnig, U. Cetintemel, : DBPal: A learned NL-interface for databases. In: Proceedings of the 2018 International Conference on Management of Data, pp. 1765–1768. ACM (2018)

[1] Trecho de "Palavras ao Vento" de Marisa Monte e Moraes Moreira

L. Blunschi, C. Jossen, D. Kossmann, M. Mori, J. Stockinger: SODA: Generating SQL for Business Users. Proc. VLDB Endow. 5(10): 932-943 (2012)

E. F. Codd: Seven Steps to Rendezvous with the Casual User. IFIP Working Conference Data Base Management 1974: 179-200

H. Kim, B. So, W. Han, H. Lee:Natural language to SQL: Where are we today? Proc. VLDB Endow. 13(10): 1737-1750 (2020)

F. Li and H. V. Jagadish. Constructing an interactive natural language interface for relational databases. PVLDB, 8(1):73–84, 2014.

P. de Oliveira, A. da Silva, E. Moura: Ranking Candidate Networks of relations to improve keyword search over relational databases. ICDE 2015: 399-410

P. de Oliveira, A. da Silva, E. Moura, R. Rodrigues: Match-Based Candidate Network Generation for Keyword Queries over Relational Databases. ICDE 2018: 1344-1347

P. de Oliveira, A. da Silva, E. Moura, R. Rodrigues:, "Efficient Match-Based Candidate Network Generation for Keyword Queries over Relational Databases," in IEEE Transactions on Knowledge and Data Engineering, To appear

F. Ozcan, A. Quamar, J. Sen, C. Lei, V. Efthymiou: State of the Art and Open Challenges in Natural Language Interfaces to Data. SIGMOD Conference 2020: 2629-2636

A. Popescu, A. Armanasu, O. Etzioni, D. Ko, and A. Yates. Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In COLING, 2004

D. Saha, A. Floratou, K. Sankaranarayanan, K., U. F. Minhas, A. R , Mittal, F. Özcan, (2016). ATHENA: an ontology-driven system for natural language querying over relational data stores. Proceedings of the VLDB Endowment, 9(12), 1209-1220.

J. Sen, C. Lei, A. Quamar, F. Özcan, V. Efthymiou, A. Dalmia, G. Stager, A. R. Mittal, Diptikalyan Saha, Karthik Sankaranarayanan: ATHENA++: Natural Language Querying for Complex Nested SQL Queries. Proc. VLDB Endow. 13(11): 2747-2759 (2020)

N. Weir, P. Utama, A. Galakatos, A. Crotty, A. Ilkhechi, S. Ramaswamy, R. Bhushan, N/ Geisler, B. Hättasch, S. Eger, U. Çetintemel, C. Binnig: DBPal: A Fully Pluggable NL2SQL Training Pipeline. SIGMOD Conference 2020: 2347-2361

X. Xu, C. Liu, and D. Song. Sqlnet: Generating structured queries from natural language without reinforcement learning. CoRR, abs/1711.04436, 2017.

V. Zhong, C. Xiong, and R. Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. CoRR, abs/1709.00103, 2017.