Evandro Eduardo Seron Ruiz
Tiago Timponi Torrent

STIL 2021

XIII Brazilian Symposium in Information and Human Language Technology and Collocated Events

Proceedings of the Conference

Online event from
November 29th to December 3rd, 2021

Editors' addresses:

Departamento de Computação e Matemática
Programa de Pós-Graduação em Computação Aplicada
Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto
Universidade de São Paulo
evandro@usp.b

Departamento de Letras
Programa de Pós-Graduação em Linguística
Universidade Federal de Juiz de Fora
tiago.torrent@ufjf.br

XIII Brazilian Symposium in Information and Human Language Technology

STIL is the bi-annual Language Technology event supported by the Brazilian Computer Society (SBC) and by the Brazilian Special Interest Group on Natural Language Processing (CE-PLN).

In 2021, STIL will be held as an online event collocated with BRACIS 2021 (The 10th Brazilian Conference on Intelligent Systems), ENIAC 2021 (The 18th National Meeting on Artificial and Computational Intelligence), and KDD-BR2021 (the 5th Brazilian Competition on Knowledge Discovery In Databases).

STIL will feature the following collocated events:

- VII Workshop on Portuguese Description (JDP); the
- VII Student Workshop on Information and Human Language Technology (TILic); and the
- OpenCor 2021 - Latin American and Iberian Languages Open Corpora Forum (OpenCor).

The conference is multidisciplinary and covers a broad spectrum of disciplines related to Human Language Technology, such as Linguistics, Computer Science, Psycholinguistics, and Information Science. It aims at bringing together both academic and industrial participants working on those areas.

We received 45 submissions. Each paper was reviewed by at least two members of the Program Committee, which had 58 members from 8 countries and various institutions. After a rigorous reviewing process, 31 papers were selected for oral presentation. We thank the authors for their submissions, the program committee for their hard work, invited speakers, SBC staff, and the BRACIS chair.

November, 2021

Evandro Eduardo Seron Ruiz
Tiago Timponi Torrent

Acknowledgments

The Program Committee chairs acknowledge the financial support to the conference provided by the Brazilian Computer Society (SBC) and the sponsors: Americanas S.A., Banco Itaú, NVidia, Ambev Tech, Google and Loggi. We thank the Program Committees of the XIII Brazilian Symposium in Information and Human Language Technology and Collocated Events for their reviews. Last but not least, we are grateful to Reinaldo Bianchi, BRACIS 2021 General Chair.

November, 2021

Evandro Eduardo Seron Ruiz
Tiago Timponi Torrent

Program chairs

Evandro Eduardo Seron Ruiz (Universidade de São Paulo)
Tiago Timponi Torrent (Universidade Federal de Juiz de Fora)

Program Committee

Alessandra Alaniz Macedo, FFCLRP–USP
Alexandre Rademaker, IBM Research
Aline Evers, UFRGS
Andre Adami, Universidade de Caxias do Sul
Ariani Di Felippo, UFSCar
Arnaldo Candido Junior, UTFPR
Arthur Lorenzi, Universidade Federal de Juiz de Fora
Bento DiasDaSilva, UNESP
Carlos Prolo, Universidade Federal do Rio Grande do Norte
Carlos Ramisch, Aix Marseille Université
Cassia Trojahn dos Santos, IRIT & UTM2
Christopher Shulby, University of São Paulo
Clarissa Xavier, UFRGS
Claudia Barros, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP
Daniela Barreiro Claro, Federal University of Bahia
Diana Santos, Linguateca, Universidade de Oslo
Diego Amancio, USP
Diogo Cortiz, PUC–SP
Ely Matos, Universidade Federal de Juiz de Fora
Eraldo Fernandes, Universidade Federal de Mato Grosso do Sul
Eric Laporte, Université Gustave Eiffel
Erick Fonseca, Real Digital
Erick Maziero, Universidade Federal de Lavras
Evandro Eduardo Seron Ruiz, USP
Francis Bond, Nanyang Technological University
Geraldo Xexéo, UFRJ
Gustavo Paetzold, University of Sheffield
Helena Caseli, UFSCar
Heliana Mello, Universidade Federal de Minas Gerais
Horacio Saggion, Universitat Pompeu Fabra
Hugo Gonçalo Oliveira, Universidade de Coimbra
Isabel Trancoso, INESC-ID, IST
Ivandré Paraboni, USP Leste
Jorge Baptista, University Algarve
Leandro Mendonça de Oliveira, Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA)
Livy Real, B2W Digital/GLiC
Lucelene Lopes, USP/ICMC
Marcelo Barros Custodio, Universidade Federal de Juiz de Fora
Marcelo Finger, USP/IME
Maria das Graças Nunes, USP/ICMC

Maria José Finatto, Universidade Federal do Rio Grande do Sul
Marlo Souza, Universidade Federal da Bahia – UFBA
Mateus Machado, USP/ICMC
Nelson Neto, Federal University of Pará (UFPA)
Norton Roman, USP/EACH
Oto Vale, UFSCar
Paulo Cavalin, IBM Research Brazil
Paulo Quaresma, Universidade de Évora
Renata Vieira, PUCRS
Sergio Freitas, Universidade de Brasília
Stella Tagnin, USP
Thiago Pardo, USP/ICMC
Tiago Timponi Torrent, Universidade Federal de Juiz de Fora
Valéria Feltrim, Universidade Estadual de Maringá
Vithor Gomes Bertalan, USP
Vladia Pinheiro, Universidade de Fortaleza
Zheng Xin Yong, Brown University

Contents

Part I

Conference Papers

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Porttinari - a Large Multi-genre Treebank for Brazilian Portuguese

Thiago Alexandre Salgueiro Pardo[1], Magali Sanches Duran[1], Lucelene Lopes[1], Ariani Di Felippo[2], Norton Trevisan Roman[3], Maria das Graças Volpe Nunes[1]

[1]Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo

[2]Núcleo Interinstitucional de Linguística Computacional (NILC)
Departamento de Letras, Universidade Federal de São Carlos

[3]Escola de Artes, Ciências e Humanidades, Universidade de São Paulo

taspardo@icmc.usp.br, {magali.duran, lucelene, arianidf}@gmail.com,
norton@usp.br, gracan@icmc.usp.br

Abstract. *This paper presents the project of a large multi-genre treebank for Brazilian Portuguese, called Porttinari. We address relevant research questions in its construction and annotation, reporting the work already done. The treebank is affiliated with the "Universal Dependencies" international model, widely adopted in the area, and must be the basis for the development of state of the art tagging and parsing systems for Portuguese, as well as for conducting linguistic studies on morphosyntax and syntax for this language.*

1. Introduction

Candido Portinari was one of the greatest artists of Brazil. He was born in Brodowski (in the São Paulo state) in 1903 and passed away in 1962 (in the city of Rio de Janeiro, in the Rio de Janeiro state), contributing to Brazilian culture and representing the challenging national reality in his work. His artistic qualities were internationally recognized and, according to the *Projeto Portinari*[1] kept by his son João Candido Portinari, the *Guerra* (War) and *Paz* (Peace) panels created for the United Nations organization are his masterpiece work, considered his "most universal" paintings.

It is not a coincidence that "Porttinari" was chosen to name the initiative that we present in this paper. More than an acronym (Porttinari stands for "PORTuguese Treebank"), it reminds us of the great challenges and equally great contributions that building a large treebank may bring to the Portuguese computational processing and linguistic studies. As defined by Jurafsky and Martin (2008), a treebank is a syntactically annotated corpus, where each sentence is paired with its parse tree, which usually contains the part of speech tag of each word in the sentence and the syntactic structuring of such words, in the form of relationships (in a dependency approach) or their building blocks (the phrases, in a constituency approach).

In the area of Natural Language Processing (NLP), a treebank may be used for developing/training tools as part of speech taggers and syntactic parsers, in charge of automatically uncovering some of the first levels of linguistic structuring of running texts. Such information is useful for several NLP applications, as sentiment analysis

[1] http://www.portinari.org.br/

(where the identification of nouns that represent entities and adjectives that modify nouns is relevant to determine what is being qualified), indexing and summarization (in which noun groups may be used to represent text content), grammar checking (where the related words may have to observe grammatical constraints, as number and gender agreement inside subjects) and machine translation (where predicates and their arguments in the original language must be syntactically ordered in the target language), among many others. More traditional linguistic investigations may also benefit from treebanks, which allow studying usual sentence structuring patterns, possible arguments and adjuncts of verbs and how they happen (and the diathesis alternations) and the behavior of some part of speech tags, among several other interesting research issues.

Research on syntax and parsing in NLP is not new for Portuguese, but, compared to the resources and tools for other languages (e.g., English), Portuguese may be considered a low resource language in this frontier. In order to fulfill this gap, we have proposed the construction of the Porttinari treebank. This paper introduces the treebank and its project details, discussing the relevant issues that a large corpus construction require. Initially aiming to rival in size the English best known reference (the Penn Treebank project of Marcus et al., 1993), other Porttinari distinguishable features include its filiation to the "Universal Dependencies" (UD) international model (Nivre, 2015; Nivre at al., 2020) and its proposal to be a multi-genre corpus in order to foster robust and general-use NLP, bringing relevant linguistic discussions into light and allowing to explore recent machine learning strategies (as transfer learning).

In what follows, we present the main related work in the area for Portuguese. The treebank project is reported in Section 3. Section 4 concludes this paper.

2. Related work

The Portuguese language counts with some treebanks, as *Floresta Sintá(c)tica* (Afonso et al., 2002; Freitas et al., 2008), CINTIL-DependencyBank (Branco et al., 2011), BDCamões DependencyBank (Parts I and II)[2], CORDIAL-SIN (Carrilho and Magro, 2010) and Tycho Brahe (Sousa, 2014), among others. *Floresta Sintá(c)tica* was probably the most representative effort for this language. Particularly interesting, this treebank has a manually revised portion called Bosque (with 9,364 sentences and 210,957 tokens), mapped to UD dependency annotation (Rademaker et al., 2017). Figure 1 shows an example of an UD-annotated sentence, in which one may see the words in their original forms, the syntactic relationships among them (in the level above) and their lemmas and part of speech tags (in the levels below).

The UD model is an attempt to standardize the morphosyntactic and syntactic analyses in the area, proposing an "universal" annotation strategy for all languages, as advocated by Nivre (2015). The model has been widely adopted for tagging and parsing tasks, having a large community of researchers who discuss its issues and contribute to the constant evolution of the model. Currently, there are already over 200 treebanks for more than 100 languages, and UD has become one of the dominant models in the area. For such reasons, we have adopted it as the basis for our treebank. Besides Bosque, which is the most popular treebank for Portuguese, the UD project also makes available

[2] Available at https://portulanclarin.net/

other 3 corpora for this language[3]: PUD (with 1,000 sentences and 21,917 tokens), GSD (with 12,078 sentences and 297,938 tokens) and DHBB (that, at the moment of the writing of this paper, had no available information).

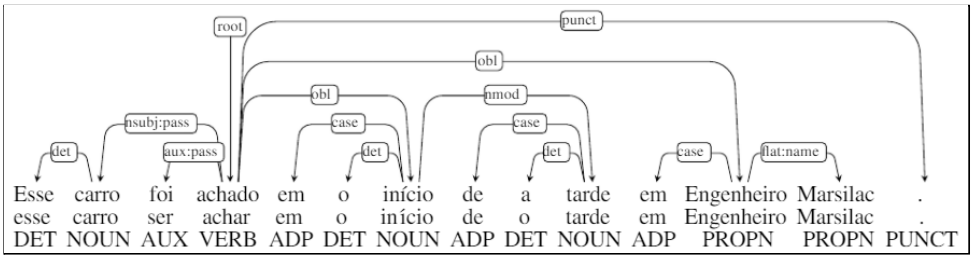


Figure 1. An example of UD-annotated sentence for Portuguese (reproduced from the work of Rademaker et al., 2017, p. 200)

Overall, the available manually (and, therefore, high quality) syntax annotated resources for Portuguese[4] are far from what other languages have access to. Consider, for instance, the English case, whose worldwide famous Penn Treebank project (Marcus et al., 1993) reports over 4.5 million words. In what follows, we report our efforts to overcome this historic limitation for the Portuguese language.

3. The rise of Porttinari

Porttinari is an ongoing initiative, aiming at growing syntax-based resources and fostering the development of related tools and applications for the Brazilian Portuguese language. The initiative started in 2020, counting with the collaboration of linguists and computer scientists, and is expected to be fully accomplished in the next few years.

Projecting the creation and annotation of a large multi-genre treebank is not a trivial task. There are several issues that must be addressed. Hovy and Lavid (2010) argue that corpus annotation is a science and that careful thought must be given to it. We present in what follows the details of the Porttinari effort according to the 7 research questions that Hovy and Lavid postulate.

3.1. Selecting the data

In order to create a multi-genre treebank, we initially selected two main genres that we consider to be more relevant to current NLP research: news texts, representing the standardized language, and the so called User-Generated Content (UGC), representing the language from web, marked by informality and produced by web users. This selection allows the study of different language genres and the creation of NLP tools and applications that may deal with varied writing styles. Another criterion for selecting material to the treebank was the license. As we expect that the treebank may subsidize other research, “open” licenses are desired, at least for the majority of the corpus.

[3] The interested reader may check them at https://universaldependencies.org/

[4] It is important to add that the other Portuguese corpora with non-UD dependency relations adopt solutions that are almost always not transferable to the UD annotation. Once we filliate to UD, we need to restrict ourselves to the set of part of speech tags and dependency relations that UD provides and to follow its guidelines.

We ended up with the following materials to be the basis for selecting the sentences that will compose the treebank, ordered from the more to the less public ones:

- news texts from *Folha de São Paulo*, from the years 2015 to 2017, made publicly available (license CC0) by Kaggle website[5] (composed by 167,053 texts);
- news texts from the MAC-MORPHO corpus, which is a 1.1 million word corpus originally developed by the Lacio-Web project (Aluísio et al., 2003), made publicly available (license CC-BY);
- stock market tweets from DANTE (Dependency-ANalised corpora of TwEets), publicly available (license GPL 2) by Silva et al. (2020), with 4,277 tweets;
- e-commerce customer reviews from B2W-reviews1 corpus, with more than 130,000 texts (license CC-BY-NC-SA), as described by Real et al. (2019);
- a small but challenging corpus of online book review sentences, as described by Belisário et al. (2020), with 350 texts (with no public license).

Such corpora sum up to nearly 80 million tokens, with almost 4 million sentences. Keeping the not fully public material (as the last two corpora in the list) is interesting for training/development (given the size of B2W-reviews1 corpus) and testing (given the difficulties of book reviews) of systems and language theories and models.

3.2. Instantiating the theory

Before starting the annotation, we investigated the UD model and built the Portuguese annotation guidelines. Although there were already three Portuguese UD corpora (PUD, GSD and Bosque), none of them released an annotation manual containing specific guidelines for Portuguese. At most, examples of the use of each part of speech tag or dependency relation in Portuguese (often examples translated from English) constituted the Portuguese (pt) tab of each item of the UD guidelines.

Directly translating guidelines may cause a detachment from UD theory. In English, for example, there are cases of two non-prepositional objects (as the dative construction "to give somebody something") that UD annotates as *obj* and *iobj*. Both *obj* and *iobj* are considered core dependents in UD, and both should be able to be promoted to subject position in passive alternation. In Portuguese, however, the recipient of the dative verbs is always prepositional and cannot be converted into a passive voice subject. These cases, according to UD, should be annotated as *obl* and not *iobj* (for example, in English, the recipient of dative verbs introduced by "to" are *obl*: to give something *to somebody*). The translation and the fact that an indirect object is understood in traditional Portuguese grammar as a prepositional object could lead us to erroneously annotate prepositional objects as *iobj*. Aware that the instantiation of theory is much more than a translation task, we have carefully analyzed each of the 17 UD part of speech tags and 37 dependency relations in order to produce Portuguese UD guidelines full of examples and clarifications about our annotation decisions.

Several theoretical challenges have arisen. Just to illustrate a few cases[6], consider the part of speech annotation. UD prescribes that an adjective should always be

[5] https://www.kaggle.com/marlesson/news-of-the-site-folhauol

[6] There are so many challenges that they will be discussed in other papers, as we have space limitations here.

annotated as *ADJ*, even though it may exceptionally head a nominal phrase. In Portuguese, unlike English, many words can be categorized as either adjective or noun, depending on the context (even dictionaries bring both options). For example, in "cuidados médicos" (medical care), "médicos" is an *ADJ*, whereas in "médicos brasileiros" (Brazilian doctors), "médicos" is a *NOUN*. Because of this, we initially adopted context-based annotation: if the word was modified by an adjective, it was considered *NOUN*; if it was modifying a *NOUN*, it was considered *ADJ*. However, after starting annotation, we realized that in some situations[7] there was no clue to decide if a word was an *ADJ* or a *NOUN*. This led us to decide to annotate as *ADJ* the adjectives occurring as *head* of a dependency relation, but never modified by another adjective. For example, the adjective "melhor" (best) is annotated as *ADJ* in "Os melhores serão recompensados" (The best [ones] will be rewarded). Other challenges come from annotating tweets. Even though Sanguinetti et al. (2020) have already proposed a unified scheme for coherent UD treatment of social media in general and for Twitter across different languages, it was necessary to define criteria for certain phenomena typical of tweets mentioning stocks from the Bovespa index. A particular case found in these tweets are the stock codes, which are usually represented by five- or six-character alpha-numerical strings, such as "Petr4" for "Petrobras" and "BBAS3" for "Banco do Brasil". These codes are so popular among investors that they are commonly used as surrogates for their company names. Because of the relevance in the domain, we annotate the stock codes as *PROPN*. One may also wonder how to deal with multiword expressions. The fact is that UD does not annotate such expressions at the part of speech level, and, at the syntactic level, it only does so for multiword expressions that have no syntactic relation between their tokens. Therefore, the expression "hot dog" is annotated as an ADJ modifying a NOUN. UD, as well, does not address light verb constructions[8], as "take advantage", which is annotated as a VERB with a NOUN as complement.

Overall, the post-annotation report of the Bosque corpus (Souza et al., 2021) was of great help, as it brings many examples of sentences that generated annotation doubts, which allowed us to foresee problems even before starting our own annotation task.

3.3. Selecting and training the annotators

As mentioned by several authors, the background and training of annotators is an open question, since some researchers claim that they should be experts and others propose training annotators just adequately for the task at hand. Given the UD annotation, the strategy used in our project to select the annotators lies between these two viewpoints. We selected 10 undergraduate students in Linguistics or Letters courses with a reasonably similar grounding in morphosyntax and syntax and offered relatively extensive training based on the Portuguese annotation guidelines.

[7] For cases like this, previous solutions proposed for Portuguese may not work. For instance, the proposal in *Bíblia Florestal* (https://www.linguateca.pt/Floresta/BibliaFlorestal), i.e., keeping the two possible tags separated by a slash, is not feasible within the UD guidelines, which only allow the assignment of one tag.
[8] However, some UD discussion issues do address the possibility of annotating multiword expressions and light verb constructions. There are two current suggestions to take these constructions into account when annotating, involving the inclusion of the related information in a new annotation level or considering it as miscellaneous/additional information.

We did 2 weeks of annotation training before starting the annotation of the corpus. During this period, virtual meetings were held 2-3 times a week, both to discuss the guidelines and to correct annotation divergences, showing the options of each annotator in the annotation tool itself (which we introduce later). From then on, the training meetings became weekly and had as theme the issues that most generated doubts in the previous week. All training meetings were recorded and the corresponding used slides made available in a common access area for the group. We also maintain at Github a issue tracking system where annotators can express their doubts and the project adjudicator (i.e., a chief linguist that is expert on the annotation theory and that evaluates and makes decisions regarding the issues) can provide explanations.

After the initial training, we did 5 weeks of blind annotation with the 10 annotators on the same data and one adjudicator. This phase was rich in the sense that we could evaluate the performance of the annotators simultaneously on the same task, as well as analyze the confusion matrix, which showed the most difficult issues. To deal with the difficulties, we started to release specific studies for such cases, including disambiguation clues and examples of use, which also resulted in improvements in the annotation guidelines.

3.4. Specifying the annotation procedure and workflow

The annotation procedure starts with the automatic annotation of all sentences considered for each corpus of the Porttinari project. This initial annotation is carried out by the UDPipe system (Straka, 2018) trained over Bosque treebank, which produces state of the art results for Portuguese under the UD model. Each set of sentences compose a Repository of Automatic Annotated Sentences (RASS) that is stored using the CoNLL-U file format, which is traditionally used in the area (it is a column-based format in which each column stores the related information of the words in the lines).

The second step is the manual revision of the sentences of each RASS that is made by picking a set of sentences for human analysis. These sets may be randomly chosen or follow some specific criteria, for example, sentences of specific sizes or sentences having interesting patterns such as target tokens or tag sequence patterns. The set of chosen sentences defines a Manual Annotation Package (MAP).

The third step is to assign MAPs to the 10 trained linguists, organized in two to three groups, with each group receiving a shuffle copy of a different MAP. Shuffling MAPs aims at avoiding bias in the annotation and possible information sharing among the annotators. An extra protection comes from the fact that each annotator does not know which other annotators are in his/her group. Each annotator confirms/corrects the annotation of the sentences using a visual annotation tool (see next subsection). The output of this step is a set of CoNLL-U files with the revised annotated sentences.

The fourth step is the adjudication of each MAP by a chief linguist to provide correct and homogeneous annotation. This is done by integrating the CoNLL-U files from each annotator with the original automatic annotation and computing agreement metrics into Package Adjudication Reports (PAR). The PARs are then analyzed and the cases with disagreements are corrected by the adjudicator.

The final step consists of the incorporation of the adjudication into CoNLL-U files that are stored in the Repository of Revised Annotation Sentences (RRAS). This procedure is repeated for as many MAPs as necessary until the selected sentences of the corpora are duly manually revised. Once each adjudication process finishes, the reports with the result of the adjudication are sent to the annotators, so that they can identify the errors and learn from them. This strategy produced different responses: the annotators who were already performing well improved even more, and the annotators who were performing less well did not improve much, perhaps because they had basic deficiencies in grammatical literacy, such as difficulty in identifying passive voice, for example.

Underlying the annotation process is the decision to separately annotate the UD levels in order to simplify each task and to produce better results for each level. This is interesting as the UD annotation has shown to be a highly sophisticated task. We started by reviewing the part of speech tags of the words in each sentence. After that, the morphologic level is semi-automatically reviewed: we use the Unitex-PB lexicon (Muniz, 2004) to retrieve the relevant morphological features of each word and then ask some human annotators to review the difficult cases and those that are not in the lexicon. Finally, the dependency relations must be fully reviewed.

3.5. Designing the annotation interface

As UD annotation is a challenging task, a good annotation interface is very important to help the annotators to clearly and easily find the relevant information and manage it, to have the necessary available functionalities (as tree visualization, searching mechanisms and editing facilities) and to guarantee that the annotated data is saved and stored. We have extended and customized the Arborator-Grew (Guibon et al., 2020) tool to include new functionalities and to correct some bugs, producing a new version of it. The new functionalities include shortcuts for faster annotation, color-based facilities for helping the annotation process, automatic checking of some UD mandatory characteristics and advanced options for project management, among others.

3.6. Choosing and applying the evaluation measures

In order to evaluate the annotation procedure, we assessed the degree to which different annotators agree on their classifications. To do so, and since we were dealing with more than two annotators, we calculate the average agreement agr_i, amongst a set of c annotators, as Artstein and Poesio (2008) propose:

$$\mathrm{agr}_i = \frac{1}{\binom{\mathbf{c}}{2}} \sum_{k \in K} \binom{\mathbf{n}_{ik}}{2} = \frac{1}{\mathbf{c}(\mathbf{c}-1)} \sum_{k \in K} \mathbf{n}_{ik}(\mathbf{n}_{ik}-1)$$

where n_{ik} represents the number of annotators, assigning the label k, from a set of K possible labels, to the same token i. As defined, agreement values range from 0%, representing total disagreement (i.e., each annotator assigns a different label), to 100% (full agreement - all annotators assigned the same label to the token). Following the authors, we take values above 80% to represent significant inter-annotator agreement, with values between 67% and 80% allowing tentative conclusions only. Such effort

helps us to evaluate how clear and reproducible the annotation task is and how annotators are understanding it, thereby increasing our confidence in the reliability of the annotation results.

Other interesting evaluation strategies are those presented by Santos and Gasperin (2002) for assessing parsed corpora, which shall be explored in the future.

3.7. Delivering and maintaining the product

The annotated portions of Porttinari must be periodically made publicly available at the project webpage. Once the treebank is ready, we also plan to make it available at the UD webpage and at the PORTULAN CLARIN portal, which is an infrastructure for research on language technologies and already includes NLP products for Portuguese.

For now, the treebank has been maintained by the efforts of the research group, which we expect to keep for the next years. Hopefully, its usefulness for the area will eventually justify its long term maintenance.

4. Final remarks

We have presented and discussed in this paper the procedures and decisions for the project of Porttinari, which shall be a large multi-genre treebank for Portuguese, affiliated with the Universal Dependencies international model. The contributions of this work include the treebank itself, the annotation process (detailed in this paper) and the theoretical issues of UD for Portuguese and the different text genres that we annotate.

So far, we have over 10,000 manually revised sentences for part of speech tagging[9], which, in sequence, were revised for lemmas and morphological features. The available data will be used to train a new tagger for Portuguese, which must produce better quality data to be reviewed, boosting the annotation process. The dependency relation revision must start in the next months.

The interested reader may find more information at the webpage of the POeTiSA project[10] (POeTiSA stands for *POrtuguese processing - Towards Syntactic Analysis and parsing*), where the related resources and tools are available (as the annotation manual and tool, the linguistic studies and the annotated portions of the corpus).

Acknowledgements

The authors are grateful to the Center for Artificial Intelligence of the University of São Paulo (C4AI - http://c4ai.inova.usp.br/), sponsored by IBM and FAPESP (grant #2019/07665-4).

Dedication

In memory of Andréia Gentil Bonfante, who very early attempted to tame the syntax and build a parser for Portuguese.

[9] Ongoing work has been confirming that this was a good decision. Since the correlation between part of speech tags and lemmas, morphological features and dependency relations is high, there has been a significant gain in the annotation.
[10] https://sites.google.com/icmc.usp.br/poetisa

References

Afonso, S.; Bick, E.; Haber, R.; Santos, D. (2002). Floresta sintá(c)tica: um treebank para o português. In Anais do XVII Encontro Nacional da Associação Portuguesa de Linguística, pp. 533-545.

Aluísio, S.M.; Pelizzoni, J.; Marchi, A.R.; Oliveira, L.; Manenti, R.; Marquiafável, V. (2003). An account of the challenge of tagging a reference corpus for brazilian portuguese. In the Proceedings of the 6th International Conference on Computational Processing of the Portuguese Language, pp. 110-117.

Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. Computational Linguistics, Vol. 34, N. 4, pp. 555-596.

Belisário, L.B.; Ferreira, L.G.; Pardo, T.A.S. (2020). Evaluating Richer Features and Varied Machine Learning Models for Subjectivity Classification of Book Review Sentences in Portuguese. Information, Vol. 11, N. 9, pp. 1-14.

Branco, A.; Castro, S.; Silva, J.; Costa, F. (2011). CINTIL DepBank Handbook: Design options for the representation of grammatical dependencies. Technical Report DI-FCUL-TR-2011-03. University of Lisbon.

Carrilho, E. and Magro, C. (2010). A anotação sintáctica do CORDIAL-SIN. In A.M. Brito, F. Silva, J. Veloso and A. Fiéis (eds.), XXV Encontro Nacional da Associação Portuguesa de Linguística. Textos seleccionados, pp. 225-241.

Freitas, C.; Rocha, P.; Bick, E. (2008). Floresta Sintá(c)tica: Bigger, Thicker and Easier. In the Proceedings of the 8th International Conference on Computational Processing of the Portuguese Language, pp. 216-219.

Guibon, G.; Courtin, M.; Gerdes, K.; Guillaume, B. (2020). When Collaborative Treebank Curation Meets Graph Grammars: Arborator With a Grew Back-End. In the Proceedings of the 12th Conference on Language Resources and Evaluation, pp. 5291-5300.

Hovy, E. and Lavid, J. (2010). Towards a 'Science' of Corpus Annotation: A New Methodological Challenge for Corpus Linguistics. International Journal of Translation, Vol. 22, N. 1, pp. 13-36.

Jurafsky, D. and Martin, J.H. (2008). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. 2a edição. Prentice Hall.

Marcus, M.P.; Santorini, B.; Marcinkiewicz, M.A. (1993). Building a large annotated corpus of English: the penn treebank. Computational Linguistics, Vol. 19, N. 2, pp. 313-330.

Muniz, M.C.M. (2004). A construção de recursos lingüístico-computacionais para o português do Brasil: o projeto de Unitex-PB. MSc Dissertation. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. 72p.

Nivre, J. (2015). Towards a Universal Grammar for Natural Language Processing. In the Proceedings of the 16th International Conference on Intelligent Text Processing and Computational Linguistics, pp. 3-16.

Nivre, J.; Marneffe, M-C.; Ginter, F.; Hajič, J.; Manning, C.D.; Pyysalo, S.; Schuster, S.; Tyers, F.; Zeman, D. (2020). Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In the Proceedings of the 12nd International Conference on Language Resources and Evaluation, pp. 4034-4043.

Rademaker, A.; Chalub, F.; Real, L.; Freitas, C.; Bick, E.; Paiva, V. (2017). Universal Dependencies for Portuguese. In the Proceedings of the 4th International Conference on Dependency Linguistics, pp. 197-206.

Real, L.; Oshiro, M.; Mafra, A. (2019). B2W-Reviews01 - An open product reviews corpus. In the Proceedings of the XII Symposium in Information and Human Language Technology, pp. 200-208.

Sanguinetti, M.; Bosco, C.; Cassidy, L.; ÇetinoĞlu, Ö.; Cignarella, A. T.; Lynn, T.; Rehbein, I.; Ruppenhofer, J.; Seddah, D.; Zeldes, A. (2020). Treebanking user-generated content: a proposal for a unified representation in universal dependencies. In the Proceedings of the 12th International Language Resources and Evaluation Conference, pp. 5240-5250.

Santos, D. and Gasperin, C. (2002). Evaluation of parsed corpora: Experiments in user-transparent and user-visible evaluation. In the Proceedings of the Third International Conference on Language Resources and Evaluation, pp. 597-604.

Silva, F.J.V.; Roman, N.T.; Carvalho, A.M.B.R. (2020). Stock market tweets annotated with emotions. Corpora, Vol. 15, N. 3, pp. 343-354.

Sousa, M.C.P (2014). O Corpus Tycho Brahe: contribuições para as humanidades digitais no Brasil. Filologia e Linguística Portuguesa, Vol. 16, pp. 53-93.

Souza, E.; Cavalcanti, T.; Silveira, A.; Evelyn, W.; Freitas, C. (2021). Diretivas e documentação de anotação UD em português (e para língua portuguesa). Available at https://nbviewer.jupyter.org/github/comcorhd/Documenta-o-UD-PT/raw/master/Documenta-o-UD-PT.pdf

Straka, M. (2018). UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In the Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 197-207.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Utilizando um dicionário morfológico para expandir a cobertura lexical de uma gramática do português no formalismo HPSG

Ana Luiza Nunes[1], **Alexandre Rademaker**[1,3], **Leonel Figueiredo de Alencar**[1,2]

[1]Escola de Matemática Aplicada da FGV (FGV/EMAp), Brasil

[2]Universidade Federal do Ceará (UFC), Brasil

[3]IBM Research, Brasil

{analuizanunes8,arademaker}@gmail.com, leonel.de.alencar@ufc.br

Abstract. *The broad lexical coverage is one of the prerequisites for the robustness of computational grammar. We propose a methodology to populate the irregular verb forms of PorGram (a Portuguese grammar in the HPSG formalism) using data from MorphoBr. We implemented an algorithm that classifies the verb forms of MorphoBr into regular and irregular, applying the inflectional rules of PorGram. We evaluated the algorithm based on a sample of 38 verbs, both regular and irregular, obtaining the expected results. An additional contribution of the work was the improvement of MorphoBr, with the elimination of more than 270,000 wrong entries and the addition of almost 13,000 missing entries.*

Resumo. *Ampla cobertura lexical constitui um dos pré-requisitos para a robustez de uma gramática computacional. Propomos uma metodologia para povoar a tabela de formas verbais irregulares da PorGram (gramática do português no formalismo HPSG) utilizando os dados do dicionário MorphoBr. Nós implementamos um algoritmo que classifica as formas verbais do MorphoBr em regulares e irregulares, aplicando as regras flexionais da PorGram. Avaliamos o algoritmo com numa amostra de 38 verbos, regulares e irregulares, obtendo os resultados esperados. Uma contribuição adicional foi o melhoramento do MorphoBr, com a eliminação de mais de 270.000 entradas erradas e a inclusão de quase 13.000 entradas faltantes.*

1. Introdução

O processamento de linguagem natural enfrenta diversos desafios atrelados à dificuldade de cobertura dos fenômenos da linguagem. A modelagem direta de regras gramaticais e entradas lexicais em um formalismo computacional tem-se revelado uma solução válida em muitos contextos que exigem uma processamento sintático profundo, não obstante os inegáveis avanços das abordagens estatísticas baseadas em dados. Um exemplo prototípico é o sistema de resolução de perguntas Watson, da IBM, que integra em sua arquitetura híbrida uma gramática do inglês baseada na modelagem do conhecimento linguístico [Ferrucci et al. 2010, McCord et al. 2012].

O formalismo HPSG (*Head-driven Phrase Structure Grammar*) [Pollard 1994, Sag et al. 2003] é um dos mais difundidos para elaboração de gramáticas computacionais de ampla cobertura. O português dispõe de uma gramática nesse formalismo, que

é a LxGram [Branco 2014, Costa and Branco 2010]. Essa gramática, porém, não é distribuída sob licença de software livre e de código aberto (doravante FOSS, do inglês *free open-source software*).

Visando a preencher essa lacuna no terreno do *parsing* do português baseado em HPSG, foi iniciado recentemente o desenvolvimento da PorGram.[1] No momento, essa gramática modela um grande número de regularidades na conjugação verbal do português por meio de 130 regras lexicais, totalizando 395 regras de reescrita. Essas regras cobrem tanto os paradigmas tradicionalmente considerados regulares quanto casos de discordância gráfica (por exemplo, *venço*, *tanjo*, *ergo*, primeira pessoa do singular do presente do indicativo de *vencer*, *tanger* e *erguer*) e alternância vocálica (*sirvo* e *durmo*, formas de *servir* e *dormir*), entre outras variações sistemáticas do padrão geral [Cunha et al. 1985]. No entanto, um grande número de verbos, como *ter*, *dar*, *querer* etc., possui diversas formas completamente idiossincráticas, não abrangidas por essas regras.

Desse modo, o objetivo deste artigo é contribuir para a construção da PorGram através do preenchimento da tabela de formas irregulares, um dos componentes de uma gramática típica no sistema LKB, o ambiente de desenvolvimento utilizado na construção da PorGram [Copestake 2002]. Para esse preenchimento serão usadas as entradas do dicionário eletrônico MorphoBr [de Alencar et al. 2018]. Uma contribuição adicional deste trabalho é a melhoria do próprio MorphoBr, uma vez que, como veremos, o algoritmo de classificação de formas verbais permitiu identificar e corrigir dezenas de milhares de erros nesse recurso.

As próximas seções apresentam o recurso lexical MorphoBr e a gramática PorGram, em seguida são descritos os passos para o preenchimento da tabela de formas irregulares. Por fim, apresentamos os resultados alcançados e as conclusões.

2. MorphoBr

O MorphoBr [de Alencar et al. 2018] é um léxico de formas plenas de alta cobertura. Abrange as formas flexionadas de substantivos, adjetivos e verbos, que são as classes gramaticais mais numerosas do português. Foi criado a partir da combinação, revisão e expansão dos dicionários eletrônicos Label-Lex [Eleutério et al. 1995] e DELAF-PB [Muniz 2004]. Incorpora, também, os neologismos gerados por [Silva 2019] por meio da aplicação de regras produtivas de formação de palavras.

Para as classes flexionáveis, as entradas do MorphoBr constituem-se de pares de forma flexionada e lema seguido de informações morfológicas, conforme a *Listing* 1. Adjetivos e substantivos totalizam mais de meio milhão de entradas cada, verbos somam mais de dois e meio milhões (não contabilizando as formas com pronomes clíticos), correspondentes a 28080 lemas.[2]

3. PorGram

Numa primeira fase, a exemplo de gramáticas análogas, como a LxGram, a PorGram está sendo implementada com o sistema *Grammar Matrix*, que permite gerar o código de uma

[1] A motivação e a estrutura dessa gramática são descritas em detalhe em artigo, de autoria do segundo e terceiro autor do presente trabalho, submetido recentemente a um periódico, ora em revisão por pares.

[2] Dados computados em versão do MorphoBr anterior às modificações descritas no presente artigo.

Listing 1. Exemplos de entradas do MorphoBr

```
1 venço   vencer+V+PRS+1+SG
2 ergo    erguer+V+PRS+1+SG
3 ajo agir+V+PRS+1+SG
4 durmo   dormir+V+PRS+1+SG
5 tenho   ter+V+PRS+1+SG
6 dou     dar+V+PRS+1+SG
```

Listing 2. Regra da terceira pessoa do plural do presente do indicativo

```
1 pres-ind-3pl-suffix :=
2 ; (partir partem) (vender vendem) (doar doam) (passear
    ↪ passeiam)
3 %suffix (ir em) (er em) (!zar !zam) (ear eiam)
4 pres-ind-3pl-lex-rule.
```

gramática inicial a partir da descrição de propriedades gramaticais da língua por meio do preenchimento de um questionário de customização [Bender et al. 2010]. Para testar a gramática, tem sido utilizado o sistema LKB, implementado em LISP [Copestake 2002].

Uma das limitações conhecidas da *Grammar Matrix* é que, no terreno da morfologia, se limita à morfotática [Goodman 2013], não permitindo modelar regras de alternância morfofonológica ou ortográfica [Beesley and Karttunen 2003]. Por exemplo, no questionário da *Grammar Matrix*, podemos modelar a formação da primeira pessoa do singular do presente do indicativo por meio da adjunção do sufixo *o* ao radical verbal. Essa regra funciona para um verbo como *comprar*. No entanto, não contempla nenhuma das formas da *Listing* 1. Enquanto as duas últimas formas discrepam idiossincraticamente do padrão conjugacional regular, as quatro primeiras exemplificam variações sistemáticas no radical verbal ou na flexão que afetam milhares de formas verbais. Desse modo, as regras codificadas inicialmente usando a *Grammar Matrix* foram modificadas manualmente, de modo a incluir o maior número possível de padrões de flexão e, assim, diminuir a quantidade de memória ocupada pela gramática.

A PorGram é codificada na linguagem *Type Description Language* (TDL), que possibilita a declaração de entradas lexicais, regras sintagmáticas e regras lexicais, como na *Listing* 2, que constitui recodificação manual de regra gerada inicialmente pela *Grammar Matrix*.

A segunda linha da *Listing* 2 é um comentário com exemplos. Na terceira linha, temos uma sequência de regras de reescrita de sufixos, que são processados sucessivamente pelo algoritmo de *parsing* do LKB [Copestake 2002, Goodman 2013]. Em cada uma dessas regras de rescrita, o primeiro sufixo constitui a entrada, enquanto o segundo constitui a saída da regra. Por exemplo, se o lema verbal termina em *ear*, a flexão deve ser *eiam*. No par de sufixos `(!zar !zam)`, `!z` é uma classe de caracteres equivalente *grosso modo* à expressão regular `[^e]`. Ou seja, *ar* só é substituído por *am* se o caractere precedente não for *e*, o que evita a hipergeração de formas agramaticais como *passeam*.

As formas irregulares, que não são geradas através das regras flexionais, precisam

Listing 3. Exemplos de entradas de formas irregulares

```
1 têm PRES-IND-3PL-SUFFIX ter
2 vão PRES-IND-3PL-SUFFIX ir
3 extinguido PAST-PART-SUFFIX extinguir
4 extinto PAST-PART-SUFFIX extinguir
```

ser listadas em uma tabela, como exemplificado na *Listing* 3. Na PorGram, essa tabela é armazenada no arquivo `my-irregs.tab`. Observe que, quando uma forma irregular coexiste com uma regular, como no caso do particípio passado de *extinguir*, ambas precisam ser listadas nesse arquivo.[3]

4. Metodologia de preenchimento da tabela de formas irregulares

Com o intuito de povoar `my-irregs.tab` a partir das entradas do MorphoBr, implementamos um algoritmo que classifica as formas verbais em regulares ou irregulares, com base nas regras flexionais da PorGram. A linguagem de programação escolhida para essa tarefa foi Haskell, que é puramente funcional e tem como vantagens a clareza e simplicidade do código.

Na PorGram, as declarações de regras lexicais, como exemplificado na *Listing* 2, são formuladas no arquivo `my-irules.tdl`. Infelizmente ainda não existe um *parser* para a leitura de arquivos TDL em Haskell. Por isso, utilizamos a biblioteca PyDelphin[4], que possibilita armazenar, em formato JSON, os atributos dos objetos `LetterSet`, que modela as classes de caracteres, e `LexicalRuleDefinition`, que modela as regras flexionais.

Dadas as diferenças de formato de representação do MorphoBr e da PorGram, criamos manualmente um arquivo contendo as etiquetas correspondentes a cada regra da gramática. Em Haskell, a partir do arquivo JSON com os atributos dos objetos `LetterSet` e `LexicalRuleDefinition`, os padrões das regras de reescrita foram codificados como expressões regulares. Devido ao grande volume de entradas verbais do MorphoBr, utilizamos uma estrutura eficiente para organizá-las, o `Data.Map`. Nele, como em um dicionário, temos chaves buscáveis e valores associados às mesmas. O `Map` das entradas tem como chave os lemas, aos quais se associam listas de tuplas em que o primeiro elemento é a forma flexionada e o segundo é a regra correspondente ao conjunto de etiquetas, quando não há uma regra correspondente o segundo elemento é uma *string* vazia.

Para obter as formas flexionadas irregulares, o algoritmo gera, para cada forma do MorphoBr contemplada por uma regra da gramática, uma nova forma através da aplicação dessa regra no lema. Por exemplo, a aplicação da regra lexical PAST-PART-SUFFIX a *extinguir*, que forma o particípio passado, produz, pelo padrão `(guir guido)`, a forma regular *extinguido*. Ao analisar a forma *extinto*, o algoritmo a compara com a que produziu, ou seja, *extinguido*, por meio da função `isRegular`

[3]Para mais detalhes sobre o funcionamento do algoritmo de *parsing* e geração morfológicos do LKB, ver [Copestake 2002].

[4]`https://github.com/delph-in/pydelphin`

e, comohttps://www.overleaf.com/project/60ef62474d10575be26f47d5 essa forma é diferente, a adiciona à tabela `my-irregs.tab`. Como vimos, nesse caso, tanto a forma irregular *extinto* quanto a regular *extinguido* são adicionadas à tabela.

A função `isRegular` tem três possíveis retornos:

1. no caso em que a forma analisada coincide com a forma produzida pela aplicação da regra, ou seja, é regular, o retorno é vazio;
2. quando a forma analisada não é igual à forma produzida pela regra, porém a forma produzida existe no MorphoBr, as duas formas são retornadas;
3. se as formas são diferentes e a forma que foi produzida pela regra não existe no MorphoBr, retorna apenas a forma analisada.

5. Resultados

O resultado esperado do algoritmo de classificação delineado acima era uma tabela de exceções com aproximadamente 10 mil entradas. Esse número era uma mera estimativa baseada na *English Resource Grammar* (ERG), aparentemente, a maior gramática implementada no formalismo HPSG [Flickinger 2000]. Nessa gramática, a tabela correspondente possui 4184 formas de verbos, correspondentes a 808 lemas verbais, ao passo que o léxico principal dessa gramática contém 4346 lemas verbais diferentes.[5]

Contrariando largamente à nossa expectativa inicial, ao executarmos o algoritmo sobre as formas verbais do MorphoBr pela primeira vez, obtivemos uma tabela com 483683 entradas. Uma análise cuidadosa desses dados indicou que o grande volume não resultava de erros de classificação do algoritmo nem de modelagem inadequada das regras flexionais. Em vez disso, foi causado por formas espúrias existentes no MorphoBr, que se subdividiam em 5 tipos principais:

1. formas no infinitivo terminadas em *á*, *ê*, *i*, *í* ou *ô*;
2. formas não imperativas na primeira ou segunda pessoa do plural não terminadas em *s*;
3. formas na segunda pessoa do singular que não do imperativo ou presente do indicativo sem terminar em *s*;
4. formas com o sufixo *ásseis* em vez de *asseis*;
5. formas com erros diversos, como *veiste* em vez de *vieste* ou *curguei* em vez de *curvei*.

A maior parte dessas formas espúrias eram duplicatas de formas corretas. Por meio de regras baseadas nesses tipos, foram eliminadas 270278 formas espúrias e incluídas 12908 entradas corrigidas. Essas correções permitiram reduzir a tabela, em nova aplicação do algoritmo de classificação, a 11581 entradas.

Uma primeira análise das entradas dessa tabela aponta para a necessidade de mais correções nas formas verbais do MorphoBr. Como podemos constatar na *Listing* 4, a segunda variante de cada uma dessas formas ou está com a ortografia desatualizada ou constitui uma forma espúria. Pelo Acordo Ortográfico da Língua Portuguesa de 2009, as formas *abóia*, *abotôo* e *adeqüemos* não se escrevem mais com diacríticos. A forma

[5] A ERG tem sido continuamente desenvolvida desde o seu lançamento há mais de duas décadas, sendo extremamente complexa. Não excluímos o fato de que possua mais entradas de verbos em bancos de dados auxiliares.

Listing 4. Exemplos da tabela final de formas irregulares

```
1  aboia    PRES-IND-3SG-SUFFIX     aboiar
2  abóia    PRES-IND-3SG-SUFFIX     aboiar
3  abotoo   PRES-IND-1SG-SUFFIX     abotoar
4  abotôo   PRES-IND-1SG-SUFFIX     abotoar
5  abstido PAST-PART-SUFFIX         abster
6  absteido         PAST-PART-SUFFIX         abster
7  adequemos        PRES-SUBJ-1PL-SUFFIX     adequar
8  adeqüemos        PRES-SUBJ-1PL-SUFFIX     adequar
9  dávamos IMPF-IND-1PL-SUFFIX      dar
10 demos    IMPF-IND-1PL-SUFFIX     dar
11 quises   FUT-SUBJ-2SG-SUFFIX     querer
12 quiseres         FUT-SUBJ-2SG-SUFFIX     querer
13 reavemos         PRES-IND-1PL-SUFFIX     reaver
14 reemos   PRES-IND-1PL-SUFFIX     reaver
```

demos não constitui forma do imperfeito do indicativo de *dar*, sendo *dávamos* a única forma correta para a primeira pessoal do plural nesse caso. Finalmente, as formas *quises* e *reemos* não integram os paradigmas de *querer* e *reaver*, respectivamente. As únicas formas corretas para as combinações de pessoa, número, tempo e modo indicados na *Listing* 4 são *quiseres* e *reavemos*.

Além de melhorias no MorphoBr, a produção da tabela também possibilitou uma revisão da cobertura das regras definidas em `my-irules.tdl`, chamando a atenção para a ausência do padrão `(er ias)` na regra `fut-pret-2sg-suffix`, que foi atualizada.

Como uma primeira forma de avaliação do algoritmo, aplicamo-lo a uma amostra de verbos altamente irregulares, constituída das entradas do MorphoBr para os verbos *dar*, *dizer*, *estar*, *fazer*, *haver*, *ir*, *poder*, *querer*, *saber*, *ser*, *ter*, *trazer*, *ver* e *vir*. Apesar de irregulares na maior parte da conjugação, esses verbos possuem algumas formas regulares. O algoritmo gerou uma tabela com as classificações esperadas para essa amostra de verbos. Por exemplo, no caso do verbo *dar*, a tabela não contém formas como *damos*, *dava* ou *daríamos*, que são geradas pelas regras flexionais e não possuem variantes irregulares no MorphoBr, apenas formas idiossincráticas como *dou*, *deu*, *déssemos* ou *déramos*, não abrangidas por essas regras.

Por outro lado, verificamos que a tabela não inclui nenhuma forma de uma amostra de verbos totalmente regulares, ou seja, cuja conjugação segue rigorosamente as regras flexionais implementadas, com exceção de variantes com ortografia anterior ao Acordo Ortográfico ou que constituem formas espúrias, como exemplificamos na *Listing* 4. Essa segunda amostra consistiu dos verbos *acuar*, *advertir*, *agir*, *atacar*, *caçar*, *chegar*, *comprar*, *distinguir*, *doar*, *dormir*, *erguer*, *ferir*, *mentir*, *partir*, *passear*, *perseguir*, *proteger*, *puir*, *ressarcir*, *seguir*, *sentir*, *vencer*, *vender* e *vestir*. Foi extraída dos comentários que documentam as diferentes regras flexionais do arquivo `my-irules.tdl`.

6. Conclusão

Neste artigo, relatamos sobre um algoritmo em Haskell para o preenchimento da tabela de formas verbais irregulares da gramática PorGram, a fim de que todo o léxico verbal do MorphoBr possa vir a ser utilizado pela gramática. Antes disso, a gramática só era capaz de analisar as formas regulares contempladas pelas regras flexionais. A primeira aplicação desse algoritmo nos mais de dois e meio milhões de formas verbais do MorphoBr resultou em uma tabela com quase meio milhão de formas, contrariando amplamente a expectativa inicial de algo em torno de 10 mil entradas.

Um exame da tabela gerada permitiu, por um lado, identificar um grupo de cinco tipos de erros no MorphoBr, levando a uma correção do recurso que eliminou 270278 formas espúrias e inseriu 12908 entradas corrigidas. Por outro lado, possibilitou constatar e corrigir um erro numa das regras flexionais. Aplicado sobre os dados atualizados tanto do MorphoBr quanto da gramática, o algoritmo de classificação produziu uma tabela com apenas 11581 entradas.

Uma avaliação manual dessa tabela constituiria um processo demorado. Por isso, optamos por uma avaliação por amostragem. Por um lado, analisamos os resultados para uma amostra de 14 verbos altamente irregulares. A tabela apresentou, nesse caso, as classificações esperadas. Por outro lado, a tabela não inclui formas indevidas de verbos regulares, a julgar pela análise dos resultados para 24 desses verbos, que instanciam os diferentes padrões de regularidade modelados pela PorGram. Desse modo, a nossa expectativa é de que a gramática agora possa analisar qualquer forma verbal do MorphoBr. Para tanto, temos ainda de corrigir as formas verbais erradas remanescentes do recurso e elaborar uma metodologia eficiente de avaliação em larga escala, dado o grande volume de dados envolvido.

Outro trabalho a ser desenvolvido num futuro próximo é revisar as regras flexionais, de modo a possivelmente incluir mais padrões regulares, e testar essas regras sob a perspectiva não apenas da análise, como fizemos aqui, mas também da geração, utilizando a capacidade geradora do algoritmo de classificação.

Referências

Beesley, K. R. and Karttunen, L. (2003). *Finite state morphology*. CSLI, Stanford, California.

Bender, E. M., Drellishak, S., Fokkens, A., Poulson, L., and Saleem, S. (2010). Grammar customization. *Research on Language & Computation*, 8(1):23–72. 10.1007/s11168-010-9070-1.

Branco, A. e. F. C. (2014). A computational grammar for deep linguistic processing of Portuguese: LXGram (version 5). Technical report, Universidade de Lisboa, Departamento de Informática.

Copestake, A. (2002). *Implementing typed feature structure grammars*. CSLI, Stanford, California.

Costa, F. and Branco, A. (2010). LXGram: A deep linguistic processing grammar for Portuguese. In Pardo, T. A. S., Branco, A., Klautau, A., Vieira, R., and de Lima, V. L. S., editors, *Computational Processing of the Portuguese Language*, pages 86–89, Berlin, Heidelberg. Springer Berlin Heidelberg.

Cunha, C., Cintra, L. F. L., et al. (1985). *Nova gramática do português contemporâneo*. Nova Fronteira Rio de Janeiro.

de Alencar, L. F., Cuconato, B., and Rademaker, A. (2018). MorphoBr: An open source large-coverage full-form lexicon for morphological analysis of Portuguese. *Texto Livre: Linguagem e Tecnologia*, 11(3):1–25.

Eleutério, S., Freire, H., Ranchhod, E., and Baptista, J. (1995). A system of electronic dictionaries of Portuguese. *Lingvisticae Investigationes*, 19(1):57–82.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefer, N., and Welty, C. (2010). Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3):59–79.

Flickinger, D. (2000). On building a more effcient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Goodman, M. W. (2013). Generation of machine-readable morphological rules with human readable input. *University of Washington Working Papers in Linguistics*, 30:1–34.

McCord, M. C., Murdock, J. W., and Boguraev, B. K. (2012). Deep parsing in Watson. *IBM Journal of Research and Development*, 56(3.4):3:1–3:15.

Muniz, M. C. M. (2004). A construção de recursos linguístico-computacionais para o português do brasil: o projeto de unitex-pb. *Master's thesis, Instituto de Ciências Matemáticas e de Computação, USP*.

Pollard, C. (1994). *Head-driven Phrase Structure Grammar*. CSLI, Chicago, Illinois.

Sag, I. A., Wasow, T., and Bender, E. M. (2003). *Syntactic theory: A formal introduction*. University of Chicago Press, Chicago, second edition edition.

Silva, H. L. B. (2019). Expansão do MorphoBr através da modelagem computacional de processos de formação de palavras em português. Master's thesis, Universidade Federal do Ceará, Brazil.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Explorando a revisão de corpora por meio da comparação de regras gramaticais em padrões sintáticos

Wellington José Leite da Silva[1], **Alexandre Rademaker**[1,3], **Leonel Figueiredo de Alencar**[1,2]

[1]Escola de Matemática Aplicada da FGV (EMAP), Brazil

[2]Universidade Federal do Ceará (UFC), Brazil

[3]IBM Research, Brazil

Abstract. *Language resources, such as corpora, are fundamental for the development of text processing tools. A resource currently considered fundamental for NLP in Portuguese is the corpus UD Bosque, part of the corpora collection in the Universal Dependencies (UD) project. Despite UD Bosque being originated from a manually revised (golden) corpus, several annotation consistency problems are encountered in its current version. In this work, we present the methodology to correct the problems of morphological annotations in the corpus; in particular, we correct morphological agreements of adjectives, determinants, and nouns. We discuss the errors, exceptions, or non-trivial cases, corrections that we made, and the impact of changes on the corpus on the training of statistical parsers.*

Resumo. *Recursos linguísticos, como corpora, são fundamentais para o desenvolvimento de ferramentas para processamento de textos. No processamento de textos em português, um recurso atualmente considerado fundamental é o corpus UD Bosque, parte da coleção de corpora no projeto 'Universal Dependencies' (UD). A despeito do corpus UD Bosque ter sido convertido para as anotações de UD de um corpus originalmente revisado, ainda são vários os problemas de consistência das anotações encontrados na atual versão do corpus. Neste trabalho, apresentamos a metodologia usada para corrigir os problemas de anotações morfológicas nos corpus UD Bosque, em particular, identificamos erros nas anotações morfológicas de determinantes e adjetivos que deveriam concordar com os substantivos que modificam. Discutimos como os erros foram identificados, as exceções ou casos não triviais, as correções realizadas e o impacto das mudanças no corpus no treinamento de analisadores sintáticos estatísticos.*

1. Introdução

O aprendizado de máquina (ML) [Mitchell et al. 1997] evoluiu do estudo de reconhecimento de padrões e da teoria do aprendizado computacional em inteligência artificial e hoje é largamente utilizado no processamento de linguagem natural (NLP). Em sua essência, tais métodos demandam dados, manualmente anotados ou não, para o treinamento de sistemas na realização de tarefas específicas.

Na linguística computacional [1] ou no processamento de linguagem natural, um analisador sintático é o primeiro componente de muitos sistemas que pretendem processar e interpretar texto. Dada uma frase como entrada, o analisador sintático marca cada palavra com uma classe gramatical (POS) e determina as relações sintáticas entre as palavras na frase. Na análise de dependências, estas relações são representadas na árvore de análise de dependência e estão diretamente relacionadas ao significado subjacente da frase em questão [Jurafsky and Martin 2009]. Um dos principais problemas que torna a análise sintática uma tarefa desafiadora é que as linguagens humanas apresentam níveis notáveis de ambiguidade. Não é incomum que frases de comprimento moderado, 20 ou 30 palavras, tenham centenas, milhares ou mesmo dezenas de milhares de possíveis estruturas sintáticas [Oepen et al. 2004]. Um analisador sintático de linguagem natural deve de alguma forma pesquisar todas essas alternativas e encontrar a estrutura mais plausível de acordo com o contexto. Os analisadores sintáticos podem utilizar uma gramática computacional implementada em formalismos como HPSG [Sag et al. 2003] ou serem baseados no aprendizado de máquina, usando grande volumes de textos (corpora), segmentados em sentenças que são associadas a sua respectiva análise sintática.

O 'Universal Dependencies' (UD) [2] é um projeto para o desenvolvimento de um esquema de anotação multilinguagem, com o objetivo de facilitar o desenvolvimento de analisador multilíngue, aprendizagem multilíngue e pesquisa na área de análise sintática com a perspectiva da tipologia linguística. O desenvolvimento de analisadores sintáticos era, antes de UD, em grande parte limitado pela falta de um grande volume de dados anotados de forma consistente seguindo um mesmo esquema de anotações (marcações e diretrizes). Isto é, vários corpora eram criados por diferentes grupos de pesquisa usando diferentes esquemas de anotação. Atualmente o projeto UD conta com três corpora em português. O corpus mais consistente e mais utilizado no treinamento dos analisadores sintáticos mais populares e livremente distribuídos é o UD Bosque [Rademaker et al. 2017].

Infelizmente, embora seja o mais importante corpus do português no projeto UD, o corpus UD Bosque tem limitações. Em primeiro lugar, é um corpus considerado pequeno, com aproximadamente nove mil sentenças apenas. Em segundo lugar, embora tenha sido convertido para o formato UD a partir de um corpus previamente elaborado em outro formalismo e manualmente revisado [Afonso et al. 2002], ainda apresenta várias inconsistências e erros de anotações, eventualmente introduzidas inadvertidamente durante suas revisões ou na conversão para o formato UD [Rademaker et al. 2017]. Motivados em resolver tais problemas, buscamos uma metodologia que nos ajude a evitar novas inconsistências e ofereça escalabilidade para a manutenção de um novo corpus para o português que deverá contar com aproximadamente 320 mil sentenças (35 vezes maior que o UD Bosque) [Ribeiro et al. 2020]. Este trabalho começou com a exploração da validação cruzada do corpus com o léxico de formas plenas do português MorphoBr [de Alencar et al. 2018]. A partir desta etapa, documentada em outro artigo de autoria dos dois últimos autores do presente trabalho, artigo esse submetido a um periódico e ainda em revisão, identificamos casos de anotações morfológicas inconsistentes de acordo com as regras da gramática do português.

[1] O termo linguística computacional é usado aqui para designar a área de pesquisa que utiliza sistemas computacionais para estudos linguísticos e se contrasta com o processamento de linguagem natural, mais aplicado, a área interessada no desenvolvimento de sistemas para processamento automático de textos.

[2] `https://universaldependencies.org`

Neste artigo, documentamos a metodologia que adotamos para correção dos erros das anotações de adjetivos e determinantes que, segundo a gramática do português, devem concordar em gênero e número com os substantivos que introduzem ou modificam. Diferentemente de uma gramática computacional, onde a regra de concordância entre adjetivos, determinantes e substantivos é explicitamente codificada, permitindo assim que um analisador sintático que utilize tal gramática seja preciso na análise de uma sentença como gramatical ou agramatical, um sistema baseado no aprendizado de máquina deve aprender as regras de concordância do português a partir dos dados. Se os dados apresentam anotações inconsistentes, espera-se que o analisador sintático não será capaz de aprender corretamente a regra de concordância.

Este trabalho está organizado da seguinte forma. Na seção 2 apresentamos o projeto UD e o corpus Bosque. Na seção 3 discutimos os erros encontrados no corpus diretamente relacionados à concordância de marcações morfológicas, indiretamente relacionados ao desvio de concordância entre palavras ou relativos a erros de concordância nos textos originais. Na seção 4, avaliamos o impacto das mudanças descritas neste trabalho no treinamento de dois analisadores sintáticos. Finalmente, apresentamos nossas conclusões na seção 5.

2. Universal Dependencies e o corpus UD Bosque

As 'Universal Dependencies' (UD) [de Marneffe et al. 2021a] são um esquema de anotação morfossintática usado para criar corpora para mais de 100 idiomas. O UD especifica um conjunto de etiquetas (tagset) e diretrizes de uso destas etiquetas para a codificação de análises sintáticas de sentenças. As relações gramaticais entre palavras são usadas para explicar como as estruturas de argumento-predicado são codificadas morfossintaticamente em diferentes idiomas, enquanto as características morfológicas e as classes gramaticais fornecem as propriedades das palavras. [de Marneffe et al. 2021b] sustenta que esta teoria é uma boa base para a anotação consistente de línguas tipologicamente diversas de uma forma a apoiar a implementação computacional da linguagem natural, bem como estudos linguísticos mais amplos. O projeto é um esforço de uma comunidade aberta com mais de 300 contribuidores, que produziram quase 200 treebanks em mais de 100 idiomas.

O UD teve sua primeira versão de corpus do português em 2015 e hoje conta com 3 corpora em português (UD Bosque, GSD e PUD), sendo o Bosque [Rademaker et al. 2017] o mais usado para treinamento de analisadores sintáticos, com cerca de 210 mil tokens e 9,3 mil sentenças. Tudo que será apresentado neste trabalho foi realizado no corpus UD Bosque, mas será, em trabalhos futuros, aplicado nos demais corpora do português.

As anotações do UD são armazenadas em arquivos CoNLL-U [3] em que sentenças são codificadas com um token por linha e cada linha com 10 colunas, a saber, (i) o índice do token na sentença (ID), (ii) a forma original da palavra na sentença ou pontuação (form), (iii) o lema ou radical da palavra (lemma), (iv) a classe gramatical (UPOS), (v) a classe gramatical específica do idioma (XPOS), (vi) a lista de características morfológicas como gênero, número e flexões verbais (feats), (vii) o id do token governante na estrutura

[3] `http://universaldependencies.org/format.html`

de dependências (HEAD), (viii) a relação de dependência com o token governante (DEPREL), (ix) o gráfico de dependências expandido (DEPS) e (x) outras anotações (MISC). Na Figura 1 temos um exemplo da estrutura.

Figura 1. exemplo de anotações morfossintáticas de uma sentença do corpus UD Bosque no formato CoNLL-U

```
# text = Maradona negou veementemente as críticas da mãe de Franco.
# sent_id = CF388-2
1   Maradona        Maradona        PROPN   _   Gender=Masc|Number=Sing                          2   nsubj   _   _
2   negou           negar           VERB    _   Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin   0   root    _   _
3   veementemente   veementemente   ADV     _   _                                               2   advmod  _   _
4   as              o               DET     _   Definite=Def|Gender=Fem|Number=Plur|PronType=Art   5   det     _   _
5   críticas        crítica         NOUN    _   Gender=Fem|Number=Plur                           2   obj     _   _
6-7 da              _               _       _   _                                               _   _       _   _
6   de              de              ADP     _   _                                               8   case    _   _
7   a               o               DET     _   Definite=Def|Gender=Fem|Number=Sing|PronType=Art   8   det     _   _
8   mãe             mãe             NOUN    _   Gender=Fem|Number=Sing                           5   nmod    _   _
9   de              de              ADP     _   _                                               10  case    _   _
10  Franco          Franco          PROPN   _   Gender=Masc|Number=Sing                          8   nmod    _   SpaceAfter=No
11  .               .               PUNCT   _   _                                               2   punct   _   _
```

3. Erros de concordância no UD Bosque

Em trabalho anterior dos dois últimos autores deste artigo,[4] a comparação do dicionário de formas plenas Morpho-Br [de Alencar et al. 2018] com o corpus UD Bosque [Rademaker et al. 2017] foi usada para verificação de inconsistências e omissões no primeiro. Durante a comparação dos recursos, a inspeção de análises morfológicas incompletas no corpus revelou também inconsistências entre anotações no próprio corpus que ignoravam as regras de concordância do português. Dentre os casos encontrados decidimos concentrar nossa análise nos artigos e adjetivos que devem concordar com os substantivos que introduzem e modificam:

> ...quer seja definido ou indefinido, o artigo caracteriza-se por ser a palavra que introduz o substantivo indicando-lhe o gênero e número. [Cunha and Cintra 1985, página 225]

> ...o adjetivo toma a forma de singular ou plural do substantivo que ele qualifica. ...o substantivo tem sempre um gênero, o que não ocorre com o adjetivo, que assume o gênero do substantivo que ele qualifica. [Cunha and Cintra 1985, páginas 264-265]

Em UD, os artigos pertencem à classe dos determinantes, termo difundido sobretudo pela gramática gerativa de Chomsky. É uma noção distribucional, como está claro na própria definição das dependências universais.[5] Pelo critério distribucional, são determinantes em português também os pronomes demonstrativos, além dos artigos, entre outras classes de palavras. Estendemos assim nossa verificação para todos os determinantes além de apenas artigos.

Para identificar os casos de desvio de concordância, utilizamos a biblioteca Udapi [Popel et al. 2017]. A consulta 1 pesquisa por tokens anotados como adjetivos (campo `UPOSTAG` com valor `ADJ`) cujo token governante seja um substantivo (`NOUN`) e cujas etiquetas morfológicas de número e gênero ou não estejam especificadas ou sejam diferentes do seu token (substantivo) governante. Em UD, a relação de dependência `amod` é usada para ligar um adjetivo ao substantivo ou pronome que ele modifica de forma composicional ou idiomática.[6]

[4] O trabalho foi submetido para publicação e encontra-se em avaliação.

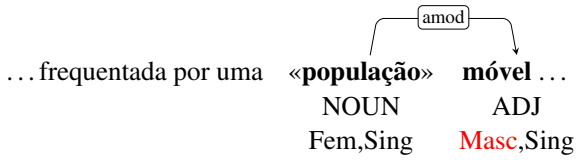
[5] `https://universaldependencies.org/u/pos/DET.html`

[6] `https://universaldependencies.org/u/dep/amod.html`

Listing 1. Exemplo de consulta no corpus usando Udapi

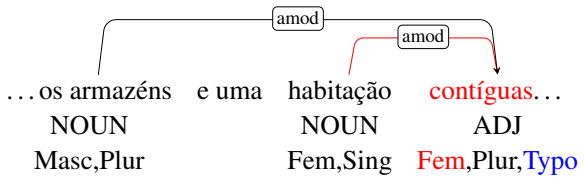
```
if ((node.feats["Gender"] != node.parent.feats["Gender"]
     or node.feats["Number"] != node.parent.feats["Number"]
     or node.feats["Gender"] == ""
     or node.feats["Number"] == "")
    and node.upos == "ADJ" and node.deprel == "amod"
    and node.parent.upos == "NOUN"):
 print(node)
```

A consulta 1, quando submetida ao corpus UD Bosque, retornou 191 casos suspeitos de falta de concordância. Os casos mais simples foram trivialmente corrigidos. No exemplo 1, o adjetivo uniforme 'móvel' [Cunha and Cintra 1985] teve a marcação de gênero corrigida de `Masc` para `Fem`.[7]



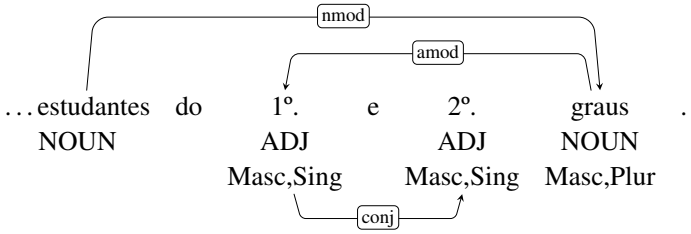
Example 1. CP52-12

Também identificamos erros gramaticais cometidos pelos autores dos textos. No exemplo 2, o adjetivo 'contíguas' deveria ter sido flexionado no masculino plural. Neste caso, UD determina que a marcação `Typo=Yes` deve ser incluída nas 'features' do token e no campo `misc`, a indicação da forma correta com `CorrectForm=contíguos`. Nota-se ainda que o erro de concordância revelou um erro na anotação sintática onde o governante do adjetivo deveria ser 'armazém', o token 'head' da coordenação.



Example 2. CP103-2

Embora a concordância de um adjetivo com uma coordenação de substantivos seja detalhadamente descrita em [Cunha and Cintra 1985], o mesmo não ocorre para construções como a apresentada no exemplo 3. Este caso consideramos como um falso positivo de nossa consulta (um falso erro) dado que entendemos a construção como gramatical.



Example 3. CF66-4

[7]Nos exemplos, marcações em vermelho sinalizam os erros encontrados cuja correção é descrita no texto, marcações introduzidas estão em azul. Os números `CFXX-XX` ou `CPXX-XX` são identificadores das sentenças do corpus.

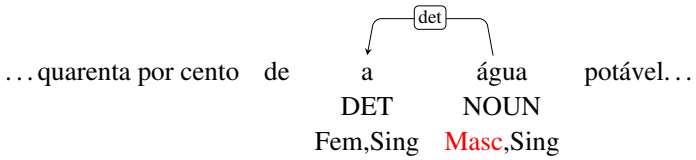
Nota-se que a consulta 1, embora produtiva na identificação dos erros mais frequentes de anotação, listou falsos erros, os quais tiveram que ser analisados manualmente. Os principais falsos erros foram casos de coordenação entre substantivos modificada por um adjetivo ou como no exemplo anterior, coordenação de adjetivos modificando um substantivo. No futuro, pretendemos explorar outros padrões como as construções de adjetivos em função predicativa. Não obstante, alguns casos não triviais também foram revelados. Em "...muitos senadores **antiaborto**..." (CP786-2) temos a palavra 'antiaborto' anotada como adjetivo e sua flexão não acompanha 'senadores', o substantivo que modifica, revelando uma possível exceção à regra de concordância do português ou uma possível necessidade de anotação mais elaborada onde apenas o afixo 'anti' seria o adjetivo (compare com 'senadores contrários ao aborto') ou seria tratado como preposição (compare com 'senadores contra o aborto'). Em "...interpretadas por bandas **cover**..." (CF840-5) temos um termo estrangeiro que optamos por anotar com a morfologia adequada à concordância, embora adjetivos no inglês não flexionem.

Para tratar os desvios de concordância entre determinante e substantivos, partimos da consulta 2, uma variação simples da consulta 1. Encontramos 1226 casos de desvios de concordância entre determinantes e substantivos. Diferentemente dos casos de concordância com adjetivos, neste caso foram possíveis algumas correções em lote. Restringindo a consulta 2 para os artigos definidos do português sem anotações morfológicas de gênero ou número, encontramos um número expressivo de casos. Como os artigos definidos do português têm gênero e número regulares, fizemos um script para adicionar as anotações morfológicas em todos os artigos definidos sem anotações morfológicas de gênero e número. Com isto, o número de casos da consulta 2 diminuiu para 282.

Listing 2. Consulta no corpus de DET/NOUN usando Udapi

```
if ((node.feats["Gender"] == ""
     or node.feats["Number"] == ""
     or node.feats["Gender"] != node.parent.feats["Gender"]
     or node.feats["Number"] != node.parent.feats["Number"])
     and node.upos == "DET" and node.deprel == "det"
     and node.parent.upos == "NOUN"):
  print(node)
```

No exemplo 4, temos um caso de simples correção, onde o token 'água' estava erroneamente[8] com gênero `Masc`.

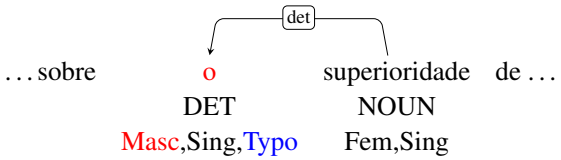


Example 4. CP452-3

Casos de erros gramaticais e ortográficos também foram encontrados, como na verificação dos adjetivos. No exemplo 5, onde autor do texto deveria ter usado o artigo definido singular feminino 'a', seguindo novamente as diretrizes de UD, utilizamos as marcações de `Typo=Yes` e `CorrectForm=a` para indicar o erro do autor do texto e

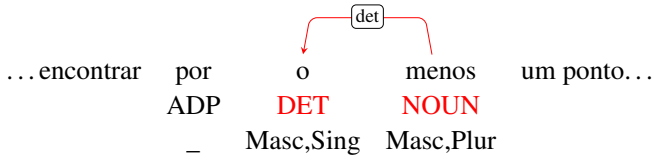
[8] E surpreendentemente para um corpus que já passou por tantas revisões.

prover uma anotação consistente para um analisador estatístico.



Example 5. CF27-7

Também encontramos casos como o exemplo 6, envolvendo a locução adverbial 'pelo menos' [9]. Neste caso, decidimos adiar o tratamento das expressões multipalavras para uma etapa futura embora nos pareça que uma análise possível seja como apresentada no exemplo 6 pelas relações de dependência.



Example 6. CF685-1

Considerando os casos aqui discutidos, no total, foram modificadas 1.094 sentenças (12% das sentenças do corpus) e 1.875 tokens (aproximadamente 1.7 tokens por sentença).

4. Avaliação

Conforme descrito na seção 1, nossa intenção é desenvolver um processo robusto e escalável para revisão de corpora. Para tal, entendemos que modificações não devem ser feitas de forma *ad hoc*. Para a avaliação do impacto de nossas alterações na qualidade do corpus, utilizamos o analisador sintático UDPipe [Straka and Straková 2017]. O UDPipe em sua versão 1.2 não é baseada em modelos de redes neurais, mas na última CoNLL 2018 Shared Task [Zeman et al. 2018] sua versão 'Future' ficou entre os três melhores colocados em todos os ranks.[10]

Os resultados obtidos são apresentados na tabela 1. Para avaliação, inicialmente, treinamos dois modelos do UDPipe com os arquivos `pt_bosque-ud-train.conllu` da versão 2.8 do corpus (de Maio 2021) e do 'branch workbench' (corrigido conforme descrito na seção 3). Usando os respectivos modelos, processamos os arquivos `pt_bosque-ud-test+dev.conllu` (juntamos os arquivos de teste e desenvolvimento que somam 1.036 sentenças) também das duas versões obtendo um arquivo `system-test+dev.conllu` para cada `pt_bosque-ud-test+dev.conllu`. Finalmente, usamos o script de avaliação desenvolvido para a CoNLL 2018 Shared Task,[11] para comparar os respectivos arquivos `pt_bosque-ud-test+dev.conllu` e `system-test+dev.conllu`. Os números da terceira coluna indicam que em todas as métricas usadas na Shared Task, o sistema melhorou seu resultado com a nova versão dos corpus.

[9] O token 'pelo' é a contração da preposição 'por' com o pronome 'o'.
[10] `https://universaldependencies.org/conll18/results.html`
[11] `https://universaldependencies.org/conll18/evaluation.html`

Existem várias métricas para quantificar a diferença entre anotações sintáticas de dependências, geralmente usadas para avaliar quão próximo é o resultado de um sistema em relação às anotações humanas ('golden') do mesmo dado. Diferentes métricas de avaliação avaliam diferentes aspectos das anotações. Na tabela 1 são apresentadas as três métricas principais usadas pelo script de avaliação da CoNLL 2018 Shared Task. As métricas LAS, MLAS e BLEX são métricas conhecidas para avaliação. A métrica 'labeled attachment score' (LAS) é uma métrica de avaliação padrão na análise de dependência: a porcentagem de tokens que são atribuídos ao token governante sintático correto e ao rótulo de dependência correto. A métrica 'Morphology-Aware Labeled Attachment Score' (MLAS) é uma extensão do CLAS (publicado experimentalmente em 2017), combinada com a avaliação de marcações de PoS tags e marcações morfológicas. A parte central é idêntica ao LAS descrito acima mas, ao contrário do LAS, certos tipos de relações não são avaliados diretamente. Palavras anexadas por meio de tais relações não são contadas como palavras independentes, sendo tratadas como características das palavras de conteúdo a que pertencem. Finalmente, a 'Bilexical dependency score' (BLEX) é semelhante ao MLAS no sentido de que se concentra nas relações entre as palavras do conteúdo. Em vez de anotações morfológicas, incorpora a lematização na avaliação. Ele está, portanto, mais próximo do conteúdo semântico e avalia dois aspectos da anotação UD que são importantes para a compreensão da linguagem: dependências e lemas.

Tabela 1. Comparação das métricas de avaliação entre as versões antiga e nova do corpus

	2.8	workbench	diferença
LAS	81.90	82.66	0.76
MLAS	67.08	67.74	0.66
BLEX	70.60	71.26	0.66

Analisadores sintáticos mais modernos têm utilizado cada vez mais pipelines baseados em redes neurais e modelos neurais pré-treinados. Um destes sistemas é o Stanza [Qi et al. 2020], a reimplementação em Python da biblioteca CoreNLP [Manning et al. 2014] de Stanford. Infelizmente, o treinamento do Stanza não terminou a tempo para que pudéssemos realizar uma comparação com a avaliação feita com o UDPipe. No entanto, na tabela 2 apresentamos a comparação entre os números parciais produzidos durante o treinamento do Stanza e os números publicados pelos autores do Stanza para os modelos pré-treinados (versão 2.5 do corpus de novembro de 2019). Obviamente esta diferença não pode ser diretamente comparada com os resultados do UDPipe, que comparam a versão atual com a versão 2.8 do corpus, que já acumula várias melhorias em relação à versão 2.5.

Tabela 2. Comparação entre os números parciais durante treinamento do Stanza e números publicados no site para o UD Bosque release 2.5

	2.5	parciais	diferença
LAS	87.57	91.11	3.54
MLAS	76.78	86.05	9.27
BLEX	80.3	87.12	7.09

5. Conclusão

Neste artigo apresentamos os primeiro passos para uma metodologia de revisão de um corpus motivada pela formalização de regras gramaticais em consultas por padrões no corpus. Em particular, revisamos o corpus UD Bosque segundo a concordância de gênero e número dos adjetivos e determinantes com os substantivos que modificam e introduzem. Conseguimos demonstrar, pela avaliação realizada, que anotações morfológicas são efetivamente usadas pelos analisadores sintáticos existentes e que houve melhora efetiva na qualidade dos dados após nossas revisões. Em trabalhos futuros, pretendemos expandir e refinar as consultas explorando talvez abordagens complementares como de [Passos 2018]. Também vale destacar métodos ligados à avaliação extrínseca dos dados, em tarefas aplicadas, como reportado em [Iwamoto et al. 2021].

Referências

Afonso, S., Bick, E., Haber, R., and Santos, D. (2002). Floresta sintá (c) tica: a treebank for portuguese. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*. ELRA.

Cunha, C. and Cintra, L. (1985). *Nova gramática do português contemporâneo*. LEXIKON Editora Digital ltda.

de Alencar, L. F., Cuconato, B., and Rademaker, A. (2018). Morphobr: An open source large-coverage full-form lexicon for morphological analysis of portuguese. *Texto Livre: Linguagem e Tecnologia*, 11(3):1–25.

de Marneffe, M.-C., Manning, C. D., Nivre, J., and Zeman, D. (2021a). Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

de Marneffe, M.-C., Manning, C. D., Nivre, J., and Zeman, D. (2021b). Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Iwamoto, R., Kanayama, H., Rademaker, A., and Ohko, T. (2021). A Universal Dependencies corpora maintenance methodology using downstream application. In *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*, pages 23–31, Online. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing*. Prentice-Hall, Inc., USA, 2 edition.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Mitchell, T. M. et al. (1997). *Machine learning*. McGraw-hill New York.

Oepen, S., Flickinger, D., Toutanova, K., and Manning, C. D. (2004). Lingo redwoods. *Research on Language and Computation*, 2(4):575–596.

Passos, G. P. (2018). A formal specification for syntactic annotation and its usage in corpus development and maintenance: a case study in universal dependencies. Master's thesis, Universidade Federal do Rio de Janeiro.

Popel, M., Žabokrtský, Z., and Vojtek, M. (2017). Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Depen-*

dencies (UDW 2017), pages 96–101, Gothenburg, Sweden. Association for Computational Linguistics.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A python natural language processing toolkit for many human languages. *CoRR*, abs/2003.07082.

Rademaker, A., Chalub, F., Real, L., Freitas, C., Bick, E., and de Paiva Universal Dependencies for Portuguese, V. (2017). Universal dependencies for portuguese. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling)*, pages 197–206, Pisa, Italy.

Ribeiro, L., Zulini, J. P., and Rademaker, A. (2020). The construction of a corpus from the brazilian historical-biographical dictionary. In Quaresma, P., Vieira, R., Aluísio, S., Moniz, H., Batista, F., and Gonçalves, T., editors, *Computational Processing of the Portuguese Language*, pages 109–117, Cham. Springer International Publishing.

Sag, I. A., Wasow, T., and Bender, E. M. (2003). *Syntactic Theory: a formal introduction*. University of Chicago Press, Chicago, second edition edition.

Straka, M. and Straková, J. (2017). Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Zeman, D., Hajič, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J., and Petrov, S. (2018). CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

PetroGold – Corpus padrão ouro para o domínio do petróleo

Elvis de Souza[1], **Aline Silveira**[1], **Tatiana Cavalcanti**[1], **Maria Clara Castro**[1], **Cláudia Freitas**[1]

[1] Departamento de Letras
Pontifícia Universidade Católica do Rio de Janeiro

{elvis.desouza99,silveira26aline}@gmail.com

tatiana.shc@hotmail.com, mariaclarac@outlook.com

claudiafreitas@puc-rio.br

Abstract. *This paper describes the creation of PetroGold, a gold standard treebank for the oil & gas domain. It is composed of theses, dissertations and monographs, contains 9,127 sentences (253,640 tokens) and has morphosyntactic annotation of dependencies according to the Universal Dependencies approach. We detail some of the linguistic challenges of the domain for syntactic annotation and assess the quality of the corpus through an intrinsic evaluation: using a model created by the UDPipe tool, the corpus leads to 90.65%, 88.53% and 82.88% of correct answers according to the UAS, LAS and CLAS measures, respectively.*

Resumo. *Este trabalho descreve a criação do PetroGold, um treebank padrão ouro para o domínio do óleo & gás. O material é composto por teses, dissertações e monografias, contém 9.127 frases (253.640 tokens) e conta com anotação morfossintática de dependências segundo a abordagem Universal Dependencies. Detalhamos alguns dos desafios linguísticos do domínio para a anotação sintática e verificamos a qualidade do material produzido por meio de uma avaliação intrínseca: utilizando um modelo criado pela ferramenta UDPipe, o corpus leva a 90,65%, 88,53% e 82,88% de acertos conforme as medidas UAS, LAS e CLAS, respectivamente.*

1. Introdução

Um dos requisitos para um Processamento de Linguagem Natural (PLN) eficiente é a existência de recursos linguísticos de qualidade, capazes de oferecer sustentação para as diversas etapas do processamento automático. Embora, para a língua portuguesa, seja possível contar com bons corpora anotados de diversas naturezas – os treebanks do projeto Floresta Sintá(c)tica [Freitas et al. 2008], a Coleção Dourada do HAREM [Freitas et al. 2010], o corpus Summit++ [Antonitsch et al. 2016], o PropBank-Br [Duran and Aluísio 2011] e a quantidade crescente de material para a língua portuguesa associado ao projeto Universal Dependencies (UD) [Nivre et al. 2016], por exemplo – o cenário é menos favorável quando se trata de domínios específicos.

As características linguísticas de domínios de especialidade podem variar bastante quando comparadas a textos considerados de linguagem geral, como corpora jornalísticos.

As diferenças vão muito além do vocabulário, estando presentes também no nível sintático e discursivo. Outro aspecto dependente de domínio é a identificação dos limites das unidades linguísticas frase e palavra, o que acarreta dificuldades para os sistemas de PLN treinados em corpora jornalísticos.

[Thompson et al. 2017] relatam que o desempenho de um parser treinado no Wall Street Journal Treebank tem uma queda de mais de 10% quando aplicado, sem qualquer adaptação, a um corpus do domínio biomédico. Do mesmo modo, [Cohen et al. 2017] informam que sistemas dedicados à resolução de correferência em domínio geral não têm um bom desempenho quando aplicados a um corpus composto por textos acadêmicos.

Neste artigo, apresentamos as etapas de construção do treebank PetroGold, composto por teses, dissertações e monografias (253.640 tokens) relacionadas à indústria do petróleo. A anotação segue a abordagem gramatical do projeto Universal Dependencies (UD).

2. Petrolês e PetroGold

O PetroGold é um subconjunto do Petrolês, que é simultaneamente um corpus e um projeto. Enquanto projeto, tem como objetivo facilitar buscas semânticas em documentos da área; enquanto corpus, trata-se de uma coleção de documentos de fontes públicas de referência na área do O&G, como a Petrobras e a Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP), contendo artigos, documentos acadêmicos, publicações periódicas, notas e estudos técnicos [Gomes et al. 2018].

Um treebank como o PetroGold tem utilidade para diferentes áreas. Do ponto de vista linguístico, permite pesquisas acerca de estruturas sintáticas tendo como base a língua em uso; do ponto de vista do PLN, permite a avaliação e o treinamento de sistemas de anotação sintática e serve ainda como subsídio para outras tarefas de PLN, como a extração de informação aberta [Gamallo et al. 2012].

Para construir um treebank padrão ouro, selecionamos um subconjunto do Petrolês de 19 teses e dissertações (253.640 tokens). Como o objetivo principal do projeto é viabilizar buscas semânticas, que por sua vez dependem da anotação de entidades do domínio (etapa futura do projeto), tomamos como critério para a seleção de documentos a presença de termos candidatos a entidade da área. Nesse primeiro momento, a anotação de entidades restringiu-se apenas à aplicação de um léxico de termos, sem revisão, compilado em [Evelyn 2021]. A Tabela 1 apresenta o PetroGold em termos quantitativos.

PetroGold	
Tokens	253.640
Palavras	223.707
Frases	9.127
Documentos	19

Tabela 1. Características do corpus PetroGold v1

Para a anotação morfossintática, utilizamos o framework do projeto Universal Dependencies, que contém 17 etiquetas para anotação de classes gramaticais e 37 etiquetas para relações sintáticas. Além das diretivas do projeto UD, questões específicas do tipo

de texto presente no Petrolês precisaram ser discutidas entre os anotadores para serem consistentemente aplicadas ao corpus tendo em vista nossos objetivos a médio prazo.

3. Desafios e opções linguísticas do PetroGold

Os desafios linguísticos na criação do PetroGold foram de dois tipos: pré-processamento e anotação morfossintática. Do ponto de vista do pré-processamento, os desafios da conversão dos arquivos PDF em arquivos de texto plano, mantendo as informações linguísticas relevantes para um corpus, foram discutidos em [Silveira et al. 2019]. Neste trabalho, nos concentramos sobre a etapa de segmentação de frases e palavras. Já os desafios morfossintáticos se relacionam ao fato de as diretivas do projeto Universal Dependencies serem genéricas e não contemplarem casos específicos do texto técnico-científico. Nesta seção discutimos alguns desses desafios e relatamos nossas opções linguísticas.

3.1. Pré-processamento

Na etapa inicial da segmentação, as primeiras unidades a serem definidas são as frases. A delimitação dessas unidades segue alguns critérios. O primeiro deles é que apenas ponto final, de exclamação e de interrogação são separadores de frases. Com isso, sinais de pontuação que poderiam ser entendidos como separadores – caso do ponto e vírgula e dos dois pontos, por exemplo – não foram caracterizados dessa forma, nem mesmo em seu uso mais frequente, como em listas e enumerações. Em ambos os casos, mesmo que haja uma quebra de linha decorrente de itemização, o fim da frase só acontece com o ponto final. Uma consequência dessa escolha quanto à segmentação é o alto número de coordenações, que passou a ser uma característica do corpus.

Outro critério de sentenciação é relacionado aos títulos e subtítulos de seções, em situação análoga ao caso das manchetes de jornal. Dado que, em geral, estes não apresentam ponto final, os segmentadores automáticos tendem a colocá-los juntos das frases que os precedem e/ou sucedem, sendo tratados como uma única frase. No entanto, a despeito da ausência de um ponto final, consideramos que os títulos precisam ser segmentados como frases autônomas, caso contrário não seria possível estabelecer uma relação sintática satisfatória entre as partes.

Nesta primeira versão do PetroGold, as frases cuja sentenciação automática fugia às diretivas foram eliminadas. Com essa decisão, eliminamos 10,3% da quantidade de frases que selecionamos inicialmente.

3.2. Anotação morfossintática

No que se refere à morfossintaxe, fenômenos típicos do gênero acadêmico e do domínio de óleo & gás, até então não abordados nas diretivas de anotação, precisaram de um tratamento sistemático, como é o caso das referências bibliográficas.

A anotação de referências bibliográficas apresenta duas exigências: (a) definir a relação entre os termos que constituem uma referência composta como ”McGurk et al. (1990)”, como ilustrado no exemplo (1), e (b) definir a relação entre a referência e o restante da frase, como ilustrado no exemplo (2).

(1) ***McGurk et al. (1990)**, analisando otólitos, encontraram evidências de redução no crescimento de larvas de arenque.*

(2) *Vale lembrar que essa técnica não é permitida dentro de os estuários (**FERREIRA, 2006**).*

Para lidar com a estrutura interna dos elementos que compõem as referências "McGurk et al. (1990)" e "FERREIRA, 2006", temos quatro possibilidades de anotação conforme as diretivas de UD, cada uma com implicações linguísticas diferentes: adjunto adnominal (nmod), coordenação (conj), expressão multi-palavra lexical sem sintaxe (flat) e expressão multi-palavra lexical com alguma sintaxe (compound). Escolhemos *flat*, pois contempla a ideia de que "McGurk et al." (sem o ano de publicação) e "FERREIRA, 2006" são, no contexto acadêmico de referências bibliográficas, um nome único, uma unidade que se refere a um trabalho específico, com uma sintaxe inexistente (ou ao menos uma sintaxe que não nos interessa marcar). Analisamos a relação entre "1990" e "McGurk" na frase (1) como *nmod*; quanto à frase (2), a relação sintática entre o núcleo do elemento entre parênteses, "FERREIRA", e a raiz da frase, "Vale", definimos como de *parataxis*.

Relações entre elementos nominais são muito comuns no domínio de óleo e gás. As convenções de anotação dessas estruturas em UD implicam distinções entre nomes próprios e comuns, o que pode ser extremamente difícil para não especialistas. Além disso, reconhecer qual relação sintática esses termos estabelecem entre si também é uma tarefa complicada. Apesar de a gramática UD dispor de algumas alternativas de classificação para expressões nominais e suas relações, as diretivas apresentam pontos ainda não completamente maduros, como a relação entre substantivos próprios e comuns, que pode ser do tipo adjunto nominal ou aposto, por exemplo[1]. No PetroGold, decidimos utilizar a etiqueta *nmod* (modificador nominal do tipo adjunto adnominal) para a maioria dos casos de nomes que modificam outros nomes. Para os casos que algumas gramáticas tradicionais chamam de *aposto especificativo* evitamos atribuir a etiqueta aposto pela dificuldade de decidir se estamos diante de expressões com dois núcleos – característica típica da etiqueta *appos* em UD – em expressões como "formação Cidreira". Assim, também analisamos essas expressões como adjunto adnominal (etiqueta *nmod*), tal como na Figura 1.

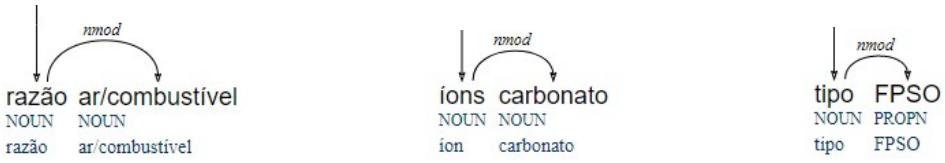


Figura 1. Anotação da relação entre nominais

Uma decisão diferente foi tomada em relação aos compostos químicos, nomes científicos e termos específicos do domínio, como "alquil glucosídeos" e "Odontesthes argentinensis", "desvio padrão" ou "efeito estufa". Todos foram anotados como *compound*.

Outras decisões de anotação do PetroGold estão descritas na documentação do

[1]Para um aprofundamento sobre as discussões, ver `https://github.com/UniversalDependencies/docs/issues/757` e `https://universaldependencies.org/workgroups/newdoc/two_nominals.html`. Acesso em 7 de ago. de 2021.

corpus.

4. Metodologia

Considerando os custos da anotação 100% manual e a capacidade de parsers atuais de produzirem resultados razoáveis, tem sido uma prática comum a criação de corpora padrão ouro por meio da revisão de material anotado automaticamente. Neste caso, o processo de revisão consiste, em grande parte, na busca por inconsistências e/ou erros da anotação automática.

No PetroGold, a revisão da anotação automática foi feita por 4 pessoas já habituadas à abordagem UD. Inicialmente, tendo em vista a familiarização com o gênero e o domínio do Petrolês, alguns documentos foram anotados por todos e as divergências discutidas em grupo. Avaliamos a concordância inter-anotadores utilizando o coeficiente κ (kappa) para cada par de anotadores. O par com concordância mais alta obteve índice de 95,1% na tarefa mais difícil, a análise das dependências sintáticas, e o par com menor concordância alcançou 91,9%.

A revisão do PetroGold foi feita sobre a saída de um modelo customizado treinado no parser Stanza [Qi et al. 2020] utilizando o Bosque-UD [Rademaker et al. 2017] v.2.6, corpus de textos jornalísticos, acrescido de um pequeno material do Petrolês. O processo de revisão demorou 3 meses, com 4 anotadores trabalhando 20 horas semanais.

Na elaboração do PetroGold, seguimos parte da estratégia de revisão utilizada em trabalhos anteriores, seguindo o que chamamos de método IAD (Inter-Annotator Disagreement) e regras linguísticas derivadas do IAD. O método IAD consiste em contrastar duas análises distintas para o mesmo corpus em busca de divergências. Contrastamos as análises fornecidas pelas ferramentas Stanza e UDPipe [Straka et al. 2016] por meio de uma matriz de confusão simplificada (Figura 2), e consideramos a ferramenta com o melhor desempenho – em nosso caso, Stanza – a anotação guia (ou anotador experiente), e UDPipe, a anotação desafiante. Trabalhamos sobre a saída da anotação guia, ou seja, se na comparação entre as duas análises a anotação guia estivesse correta, não era preciso realizar nenhuma modificação. Já se a anotação desafiante ou nenhuma das duas estivesse correta, era necessário editar o arquivo, que corresponde à saída do Stanza.

A estratégia de examinar, por meio da matriz de confusão, as divergências entre análises automáticas como potenciais casos de erro se baseia na hipótese de que se há convergência entre os anotadores, então existe acerto. A visualização das divergências pela matriz de confusão nos permite generalizar e criar hipóteses a partir dos tipos de erros – ou inconsistências – mais comuns, viabilizando a percepção de padrões nos erros e, consequentemente, a elaboração de regras capazes de detectá-los e corrigi-los de forma sistemática.

Por exemplo, a análise dos casos em que os sistemas divergiram entre a classificação de adjuntos adnominais (nmod) e adjuntos adverbiais (obl) nos possibilitou depreender um padrão: quando havia ocorrência de alguns adjetivos transitivos (como ”favoráveis”, na frase (3)), o sistema guia anotava seu complemento (”suprimento”) erroneamente como adjunto adnominal do substantivo (”condições”), e não como complemento do adjetivo (”favoráveis”).

(3) *Ainda segundo os citados autores, a sequência S4 é dominada por depósitos*

sistema	acl	acl:relcl	advcl	advmod	amod	appos	aux	aux:pass	case	cc	ccomp	c
golden												
acl	5070	8	128	0	370	12	1	0	1	0	2	1
acl:relcl	16	2246	60	0	21	4	17	1	0	0	42	0
advcl	305	73	2829	4	55	5	17	0	0	0	59	0
advmod	0	2	1	6549	24	8	3	0	236	35	1	0
amod	139	8	17	41	15970	75	4	2	19	2	12	3
appos	10	5	0	9	44	2389	0	0	15	0	1	5
aux	0	1	4	0	0	0	281	7	0	0	1	0
aux:pass	0	0	1	4	0	0	64	3590	0	0	0	0
case	0	0	3	76	10	24	0	0	45124	68	1	1
cc	0	0	4	117	0	17	0	0	49	7221	0	0
ccomp	5	28	73	5	11	0	4	2	2	0	520	0
compound	1	0	0	6	27	17	0	0	9	0	0	2
conj	69	69	132	50	113	375	20	0	35	0	47	1

Figura 2. Matriz de confusão de etiquetas sintáticas utilizada no método IAD

siliciclásticos que refletem ***condições*** *climáticas úmidas* ***favoráveis*** *ao* ***suprimento*** *sedimentar e desfavoráveis à formação de carbonatos.*

O sistema desafiante, por sua vez, costumava acertar esses casos. Assim, por um lado, a divergência serviu para nos mostrar uma questão complexa para o parser de melhor qualidade; por outro lado, nos indicou uma possível regra linguística para resolver a grande quantidade de erros desse tipo – na presença de certos tipos de adjetivo (que podemos chamar de adjetivos transitivos, como ”favorável”, ”constituinte” e ”existente”), as chances são altas de que o substantivo à direita complemente o adjetivo, e não o substantivo sendo adjetivado.

Como se trata de uma tendência, e não de uma regra determinística, criamos uma ferramenta que nos permite aplicar regras, analisar o resultado e aceitar a alteração apenas nos casos adequados ou realizar quaisquer outras modificações nas frases[2]. Para a análise das matrizes de confusão utilizamos o Julgamento, um ambiente para avaliação de corpora anotados [de Souza and Freitas 2021].

5. Resultado e análise

Os métodos utilizados na revisão do corpus resultaram na análise de 5.107 frases do PetroGold (55,9% do corpus), sendo alterada a anotação de 12.832 tokens (5,7% de todos os tokens). A Tabela 2 quantifica as correções realizadas no corpus por tipo de informação linguística considerando também as interseções (quando um token recebeu mais de uma correção). Além disso, indicamos quantos desses tokens corrigidos o método IAD, que busca por divergências na relação de dependência entre dois anotadores automáticos, identificou.

O método IAD indicou a presença de 25.123 tokens com anotação de relação de dependência divergentes. Desses, 5.656 tiveram alguma anotação corrigida. Por um lado, isso significa que mais da metade das correções realizadas no corpus não foi fruto direto

[2]Disponível em `https://github.com/alvelvis/conllu-merge-resolver`. Acesso em 7 de ago. de 2021.

Anotação	Tokens corrigidos	IAD
Qualquer correção	12.832	5.656
$LEMA$	2.258	768
POS	3.537	1.910
$HEAD$	6.780	2.865
REL	8.206	4.713
$LEMA \cap POS$	924	521
$LEMA \cap REL$	893	523
$POS \cap REL$	2.958	1.783
$HEAD \cap REL$	4.141	2.299

Tabela 2. Tipos de correção realizados no PetroGold

das matrizes de confusão (7.176, ou 55,9% das correções). No entanto, indiretamente as matrizes nos ajudaram a encontrar problemas na anotação das frases, apontando para fenômenos que precisam de atenção, como o exemplo *obl vs. nmod* apresentado na seção anterior. Esses casos foram identificados e corrigidos manualmente pelos revisores ou por meio das 25 regras de correção em lote desenvolvidas durante o processo de revisão.

Por outro lado, é interessante notar como a qualidade do anotador automático que utilizamos como sistema guia no método IAD (o Stanza) produziu resultados superiores aos do sistema desafiante (UDPipe), uma vez que, das 25.123 divergências identificadas, apenas 5.656 (22,5%) foram os tokens que precisaram de correção – nos outros 77,5% dos casos o sistema guia já estava correto. Dos tokens que foram corrigidos pelo método IAD, por sua vez, em 3.525 casos (62,3%) o sistema desafiante estava correto, enquanto que no restante dos tokens nenhum dos sistemas acertou e foi necessária uma terceira análise, humana.

Das 2.258 correções de lema, 1.247 (55,2%) não se associam a erro de qualquer outra informação linguística – são erros apenas de lema, decorrentes sobretudo da falta de familiaridade dos anotadores automáticos com as palavras do domínio, mas que, apesar da falha na lematização, não prejudicaram o parsing. Já os erros de POS parecem se associar diretamente às falhas na classificação da relação de dependência, uma vez que 83,6% dos erros de POS foram também erros de REL. O erro no encaixe de dependências sintáticas também se associa à falha na classificação da relação – 61% dos erros de HEAD também foram de REL. O contrário, no entanto, não é tão expressivo: apenas 50,4% dos erros de REL são também erros de HEAD.

Para uma avaliação intrínseca do PetroGold, separamos o corpus em partições de treino e teste e criamos um modelo utilizando o UDPipe v.1.2.0 sob os parâmetros de treinamento padrões da ferramenta. Como contraste, treinamos também um modelo a partir do Bosque-UD v.2.8, de textos jornalísticos, utilizando os mesmos parâmetros e garantindo a mesma distribuição de frases nas partições de treino e teste, seguindo a proporção de 95% e 5%, respectivamente, resultando em 8.671 frases para treino no PetroGold e 8.328 no Bosque-UD. São, portanto, dois corpora muito próximos em tamanho. Os resultados estão na Tabela 3.

As métricas de avaliação são as do CoNLL 2018 Shared Task [Zeman et al. 2018], onde UPOS avalia os acertos de classe gramatical, UAS avalia os acertos de encaixe

	LEMA (%)	UPOS (%)	UAS (%)	LAS (%)	CLAS (%)
PetroGold	98,48	98,19	90,65	88,53	82,96
Bosque-UD v.2.8	96,95	96,52	85,83	81,59	73,80

Tabela 3. Avaliação intrínseca de modelos treinados no UDPipe

de dependências sintáticas, LAS avalia a classificação das relações de dependência que foram corretamente encaixadas, e CLAS os acertos de LAS para as palavras consideradas de "conteúdo"[3].

Como podemos observar, o modelo treinado a partir do PetroGold apresentou desempenho significativamente superior. A diferença é de aproximadamente 1,5% para a lematização, mais de 1,5% para a anotação de classes gramaticais, mais de 4% no encaixe de dependências, 7% na classificação das relações e 9% na classificação das relações para palavras de conteúdo. Isso indica um grau de consistência interna maior na anotação do PetroGold, já que o modelo parece ter generalizado melhor durante o aprendizado, embora não seja possível estabelecer comparações diretas entre ambos os corpora. O gênero acadêmico, que pode ser bastante formulaico e previsível, também pode ter contribuído para os números mais altos do PetroGold.

6. Considerações finais

Apresentamos a primeira versão do PetroGold, um treebank padrão ouro para o domínio do petróleo. A intenção do material é servir como subsídio para o desenvolvimento de ferramentas de PLN específicas deste domínio, seja como material para treinamento de novos modelos ou para sua avaliação. O corpus está disponível na página do projeto Petrolês[4] e integrará o acervo do projeto Universal Dependencies.

Neste trabalho discutimos os desafios linguísticos relativos ao pré-processamento e à anotação morfossintática específicos de textos técnico-científicos do domínio do qual o Petrolês faz parte. Embora as questões linguísticas sejam específicas de um domínio, o procedimento de identificar as dificuldades no processamento automático e a forma como as resolvemos independem do tipo de texto.

Sugerimos ainda um caminho promissor para a revisão de corpora previamente anotados por ferramentas de PLN, resultando na revisão de mais de 50% das frases do corpus. Por fim, indicamos também a qualidade do material a partir da avaliação intrínseca de um modelo treinado a partir dele, chegando a 98% de acerto de classes gramaticais e 88% de acertos de dependências sintáticas.

Agradecimentos

Este trabalho foi financiado com o apoio da ANP – Agência Nacional de Petróleo, Gás Natural e Biocombustíveis, Brasil, associado ao investimento de recursos oriundos das Cláusulas de P, D & I, por meio de Termo de Cooperação entre a Petrobras e a PUC-Rio.

[3]As siglas significam, respectivamente, do inglês, Universal Part-of-speech Score, Unlabeled Attachment Score, Labeled Attachment Score e Content-Word Labeled Attachment Score.

[4]`https://petroles.puc-rio.ai`. Acesso em 7 de ago. de 2021.

Agradecemos à equipe do Laboratório de Inteligência Computacional Aplicada da PUC-Rio pela geração de modelos de anotação morfossintática customizados, e Elvis de Souza agradece ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa de mestrado de processo nº 130495/2021-2.

Referências

Antonitsch, A., Figueira, A., Amaral, D., Fonseca, E., Vieira, R., and Collovini, S. (2016). Summ-it++: an enriched version of the summ-it corpus. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2047–2051, Paris, France. European Language Resources Association (ELRA).

Cohen, K. B., Verspoor, K., Fort, K., Funk, C., Bada, M., Palmer, M., and Hunter, L. E. (2017). The colorado richly annotated full text (craft) corpus: Multi-model annotation in the biomedical domain. In *Handbook of Linguistic Annotation*, pages 1379–1394. Springer.

de Souza, E. and Freitas, C. (2021). Et: A workstation for querying, editing and evaluating annotated corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online. Association for Computational Linguistics.

Duran, M. S. and Aluísio, S. (2011). Propbank-br: a brazilian portuguese corpus annotated with semantic role labels. In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*.

Evelyn, W. F. D. (2021). Dos termos às entidades no domínio de petróleo. Master's thesis, PPGEL/PUC-Rio.

Freitas, C., Carvalho, P., Oliveira, H. G., Mota, C., and Santos, D. (2010). Second HAREM: advancing the state of the art of named entity recognition in Portuguese. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2010)*, pages 3630–3637. European Language Resources Association.

Freitas, C., Rocha, P., and Bick, E. (2008). Um mundo novo na floresta sintá (c) tica–o treebank do português. *Calidoscópio*, 6(3):142–148.

Gamallo, P., Garcia, M., and Fernández-Lanza, S. (2012). Dependency-based open information extraction. In *Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP*, pages 10–18.

Gomes, D., Cordeiro, F., and Evsukoff, A. (2018). Word embeddings em português para o domínio específico de óleo e gás. In *Proceedings of the 19th Rio oil & gas expo and conference*, page 10.

Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Rademaker, A., Chalub, F., Real, L., Freitas, C., Bick, E., and de Paiva, V. (2017). Universal dependencies for portuguese. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 197–206.

Silveira, A., de Souza, E., Cavalcanti, T., and Freitas, C. (2019). Do pdf ao txt: Desafios na extração de informação em textos técnico-científicos. In *VI Workshop de Iniciação Científica em Tecnologia da Informação e da Linguagem Humana (TILic 2019)*.

Straka, M., Hajic, J., and Straková, J. (2016). Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Thompson, P., Ananiadou, S., and Tsujii, J. (2017). The genia corpus: Annotation levels and applications. In *Handbook of Linguistic Annotation*, pages 1395–1432. Springer.

Zeman, D., Hajic, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J., and Petrov, S. (2018). Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Lexicalidade biomédica e sua mensuração em um corpus sobre COVID-19 em língua portuguesa

Karhyne S. Padilha de Assis[1], Camila das Merces Silva[1], Janaína da Silva Leite[1], Wellington Araujo Nogueira[1], Kenji Nose Filho[1], André K. Takahata[1], Margarethe Steinberger-Elias[1].

[1]Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas
Universidade Federal do ABC (UFABC) - Santo André - SP - Brasil

{karhyne.assis, camila.merces, janaina.leite, wellington.araujo, kenjinose, andre.t, margarethe.elias}@ufabc.edu.br

Abstract *We analyzed the biomedical lexicon of a corpus of Portuguese texts on Covid-19 extracted from the Pubmed database with the help of classic measures like lexical density and lexical diversity. Preliminary results could not show the lexical distribution in different texts genres and clinical specialties present in the corpus. Based on the concept of "biomedical lexicality", a new indicator, Lex-BioMed, was proposed and tested demonstrating good performance.*

Resumo. *Analisamos o léxico biomédico de um corpus de textos em língua portuguesa da base Pubmed sobre a Covid-19. A adoção inicial de medidas clássicas de densidade e diversidade lexical não foi capaz de evidenciar a distribuição lexical nos diferentes gêneros e especialidades clínicas de que se compõe o corpus. Com base no conceito de "lexicalidade biomédica", foi proposto e testado um novo indicador, o Lex-BioMed, com bons resultados.*

1. Introdução e motivação

A COVID-19, inicialmente identificada em Wuhan na China em dezembro de 2019, teve aumento de casos em diversos países e continentes, até ser decretada pandêmica pela Organização Mundial da Saúde (OMS) em 11 de março de 2020 [Cucinotta e Vanelli, 2020]. A partir desta data, textos biomédicos publicados na base Pubmed foram filtrados por Leite et al. (2020) para a criação de um corpus em português a respeito da COVID-19. Com o objetivo de prover dados para uma pesquisa dos processos de simplificação da linguagem da Saúde, o corpus COVID-19 visa atender a demanda informacional de um público brasileiro leigo sobre a pandemia. Compõe-se de 254 textos do período inicial da pandemia, compreendido entre março e setembro de 2020. O corpus é heterogêneo, distribuído entre 23 gêneros textuais e 16 especialidades clínicas. Sendo o nível lexical o de complexidade mais visível no corpus, optou-se no presente trabalho pela análise do léxico biomédico e procedeu-se à contagem de *types* e *tokens* e à mensuração da diversidade ou riqueza lexical (DiL) dada pela razão entre número de *types* e *tokens* (TTR, *type-token ratio*) e à densidade lexical (DeL) dada pela razão entre o número de palavras de conteúdo semântico (nomes, adjetivos e verbos) e o número total de palavras do corpus [Santos et al., 2018].

O nosso problema inicial de pesquisa foi como identificar automaticamente termos biomédicos de difícil compreensão e convertê-los em expressões acessíveis. Tomou-se como ponto de partida a hipótese de que os índices de densidade e de diversidade lexical seriam capazes de apontar os gêneros de maior complexidade, isto é, onde haveria maior concentração de *types*. Métodos clássicos como Flesch-Kincaid, partições morfológicas e outros foram temporariamente deixados de lado, colocando-se o texto especializado como

foco da pesquisa. “Tendo-se o texto como foco, deixa de fazer sentido que se continue estudando somente os termos, de forma que se passa a englobar os modos de dizer peculiares de cada área de conhecimento” [Finatto, 2004a, p. 348].

Linguagens de especialidade, como é o caso da biomédica, tem um comportamento diferenciado nos estudos de corpora: “(...) a partir da observação da linguagem especializada em corpora que se percebe mais francamente como a observação de termos é somente um pequeno passo na observação do texto especializado” [Perna, Delgado e Finatto, 2010 p.138]. Estudo de Zilio (2009) sobre textos científicos de Cardiologia e Radiologia compara a distribuição lexical entre as duas especialidades e atribui as convergências a fatores textuais: “(...) se não houvesse no corpus de Radiologia um artigo que se ocupasse do coração, somente dois dos compostos estudados seriam comuns aos dois corpora” (p.142).

A observação inicial sobre o comportamento lexical das linguagens de especialidades nos textos do corpus e a indefinição das medidas de densidade lexical nesse contexto levou a busca de um novo indicador da lexicalidade no corpus. Propomos aqui o conceito de “lexicalidade biomédica” ou “densidade lexical biomédica” para identificar com maior segurança o espaço lexical que é das especialidades biomédicas e diferenciá-lo de um léxico fronteiriço revelado em gêneros menos técnicos. O problema de pesquisa foi revisto, tornando-se imperativo reconhecer no corpus um repertório de termos biomédicos, de modo a identificar sua distribuição no corpus sobre COVID-19. Investigamos o novo indicador de lexicalidade para verificar se seria capaz de cumprir uma função distribucional que as simples densidade e diversidade lexicais não lograram alcançar.

2. Materiais e Métodos

Como já descrito na Seção 1, seguindo [Leite et al., 2020], foi obtido um corpus de textos em língua portuguesa a respeito da COVID-19, a partir de textos indexados na base científica *Pubmed* do período entre março e setembro de 2020, totalizando 254 textos. Os textos foram categorizados manualmente conforme gêneros, utilizando informações fornecidas pela base, e conforme especialidades clínicas, de acordo com o nome da revista, título do artigo ou palavras-chave do texto. As distribuições dos textos nas classes obtidas se encontram nas Tabelas 2 e 3. Após a fase preliminar de filtragem, limpeza e compilação, buscou-se nomear os arquivos já em formato .txt, tokenizar e fazer a anotação morfossintática das classes de palavras (PoS, *parts of speech*) com o uso do analisador e corretor gramatical CoGrOO [Silva, 2013]. O resultado para as classes de palavras de conteúdo semântico (nome, verbo e adjetivo) se encontra na Tabela 1.

Com finalidade de caracterizar e descrever o léxico biomédico do corpus, foram calculados os índices da DeL e da DiL. A DeL descreve a proporção de palavras de conteúdo pelo número total de palavras em cada texto e de acordo com [Ure, 1971] [Johansson, 2008], o valor resultante desse indicador expressa a concentração de conteúdo lexical presente em um determinado texto. Um texto com alta densidade lexical contém mais palavras de conteúdo, enquanto um texto com baixa densidade lexical é composto por palavras funcionais (preposições, conjunções, artigos, pronomes, verbos modais e auxiliares). Já a DiL pode ser descrita pela TTR. Entretanto, a TTR tende a possuir valor menor em textos com maior número de *tokens* ou a possuir valor maior em textos com menor número de *tokens*, fazendo com que o seu uso torne enviesada a comparação entre dois textos ou corpora com número de *tokens* diferentes [Johansson, 2008]. Para mitigar esse efeito, utilizamos também o vocabulário teórico (VT) ou *theoretical vocabulary* [Broeder, Extra & van Hout 1986], em que a proporção de *types* é calculada para subconjuntos de tamanho fixo com *N tokens*, no nosso caso $N=100$, amostrados aleatoriamente. Em nosso trabalho,

mostramos resultados obtidos a partir da realização de S=100 amostragens aleatórias distintas para cada grupo de texto.

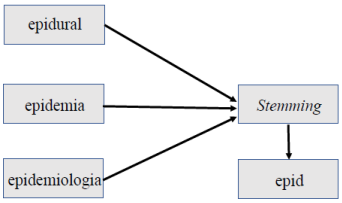


Figura 1: *Stemming*

Tabela 1: Resultados para as PoS por subcorpora

Especialidade	Nº de publicações	Total de *tokens*	n	adj	v	n-adj	n (%)	adj (%)	v (%)
Saúde Pública	55	83028	40997	11376	17707	117	49,38	15,03	21,33
Epidemiologia	36	50810	24542	7381	10267	88	48,30	14,53	20,21
Cardiologia	30	37891	16429	6690	7981	65	46,00	17,66	21,06
Enfermagem	24	46410	23265	6476	10083	68	50,13	13,09	21,73
Cirurgia	17	25098	10693	3548	5138	22	42,60	14,14	20,47
Outros	16	21446	11656	3827	5498	49	54,35	17,84	25,64
Nefrologia	14	14930	6882	2196	3122	21	46,10	14,71	20,91
Terapia Intensiva	11	15534	7424	2209	3219	20	47,79	14,22	20,72
Pneumologia	10	4791	2289	823	1000	3	47,78	17,18	20,87
Relatório Imagens	9	3717	1939	755	757	3	46,79	20,31	20,37
Não Classificados	8	6413	3062	876	1368	10	47,75	13,66	21,33
Anestesiologia	7	6672	3073	1071	1520	4	46,06	16,05	22,78
Clínica Geral	5	2733	1280	399	624	8	46,83	14,60	22,83
Fonoaudiologia	5	4131	2094	604	941	3	50,69	14,62	22,78
Pediatria	4	5679	2569	919	1158	7	45,24	17,94	20,39
Saúde Primária	3	2177	1192	330	386	3	50,16	15,16	17,73

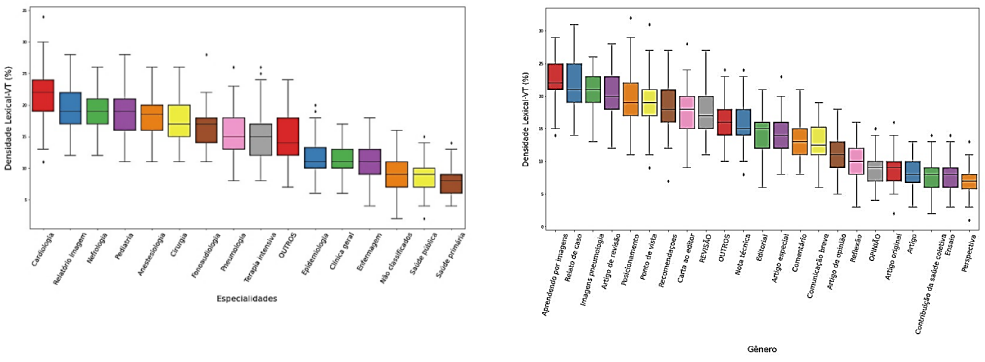
Para a identificação das palavras de domínio biomédico em toda extensão do corpus foi usado, como primeiro passo, o *stemming*, sendo empregado para implementação o RSLP Stemmer [Orengo e Huyck 2001] que tem como objetivo remover os sufixos, para reduzir o número de palavras, conforme ilustrado na Figura 1. Observamos que originalmente o corpus possui 31.123 *types*. Após o *stemming* esse número se reduziu para 11.513 raízes distintas. Após a obtenção das raízes, foi criada uma lista de pares do tipo (*raiz*, *type*), em que o segundo elemento consiste em um type escolhido ao acaso dentre as palavras do corpus que podem ser formadas com o uso da respectiva raiz. Assim, para a raiz "epid" na Figura 1, o par formado poderia ser ("epid", "epidural"). Os *types* de todos os 11.513 pares foram analisados por três pesquisadores de processamento de linguagem natural (PLN) em textos biomédicos, sendo o time formado por duas mulheres e um homem, em que dois são docentes universitários (com formação em engenharia elétrica e linguística) e uma é estudante de mestrado (com formação em análise e desenvolvimento de sistemas), com idades entre 34 e 69 anos. Procedeu-se à análise de modo a comparar a identificação de palavras com uso prevalente no domínio biomédico, resultando em 2.258 raízes associadas a palavras no domínio biomédico.

A partir dessa identificação de raízes biomédicas, passamos para a etapa de análise de cada subcorpus formado por textos agrupados por gênero ou especialidade com cálculo do DeL, DiL-TTR e DiL-VT. Constatando que tais índices não foram capazes de mostrar a distribuição lexical por gênero e por especialidade, criamos um novo indicador que pudesse medir a lexicalidade biomédica (Lex-BioMed), que consiste na proporção do número de *types* que possuem raízes biomédicas em conjuntos de tamanho fixo (no nosso caso N=100)

obtidos aleatoriamente no grupo de textos em análise. Assim como para a DiL-VT, mostramos resultados para Lex-BioMed obtidos a partir da realização de S=100 amostragens aleatórias distintas.

3. Resultados

Nas Tabelas 2 e 3 são apresentados os resultados para os subcorpora formados a partir da categorização dos textos por especialidades clínicas e por gênero textual, respectivamente. Ao analisarmos o DiL-TTR, observamos que o principal fator de influência para essa métrica foi o número de *tokens*. A influência dos números de *tokens* na medida não se observou com o DeL e com o DiL-VT, mas, como mostrado nas Tabelas 2 e 3, os valores obtidos revelaram-se similares entre os subcorpora analisados, não permitindo uma discriminação clara a respeito do conteúdo biomédico presente nos respectivos subconjuntos de textos. Contrastando com as métricas mais tradicionais, ao analisarmos a Lex-BioMed, é possível observar que se torna possível diferenciar subcorpora ou grupos de subcorpora diferentes a partir dessa métrica, o que também é corroborado pela Figura 2. Por exemplo, para especialidades, as categorias com maior Lex-BioMed foram "Cardiologia" e "Relatório de Imagens" com 25,29% e 24,57% respectivamente, enquanto as categorias de menor Lex-BioMed foram "Saúde Pública" e "Saúde Primária", com, respectivamente, 9,17% e 8,54%. Na categorização por gêneros, os de maior Lex-BioMed foram "Aprendendo por Imagens" e "Imagens Pneumologia" com proporção de 29,84% e 27,35% respectivamente. Na Figura 2, os *box-plot* indicam que a distribuição da proporção de palavras com raízes biomédicas em conjuntos de N=100 palavras escolhidas aleatoriamente é diferente entre as categorias, principalmente dentre as com maior e menor valor de Lex-BioMed. Ao ordenar as subcategorias em ordem decrescente de Lex-BioMed, como na Tabela 2 e na Figura 2(a), observamos que podemos agrupar as especialidades em dois subconjuntos. O primeiro composto por "Cardiologia", "Relatório Imagens", "Nefrologia", "Pediatria", "Anestesiologia", "Cirurgia", "Fonoaudiologia", "Pneumologia" e "Terapia Intensiva"; e o segundo formado por "Outros", "Epidemiologia", "Clínica Geral", "Enfermagem", "Não Classificados", "Saúde Pública" e "Saúde Primária". Nota-se que, em linhas gerais, no primeiro grupo encontram-se especialidades médicas mais específicas, como Cardiologia, Radiologia e Nefrologia, enquanto, no segundo grupo, encontram-se categorias indefinidas, como "Outros" e "Não Classificados", categorias não médicas, como "Enfermagem" e "Saúde Pública", ou de caráter mais amplo, como "Clínica Geral".



(a) (b)

Figura 2: *Box-plot* descrevendo a distribuição da proporção de raízes biomédicas em S=100 amostras aleatórias obtidas para cálculo de *Lex-BioMed* para cada (a) especialidade e (b) gênero textual.

Tabela 2. Resultados das características por especialidade

Especialidade	Nº publicações	Total de *tokens*	Total de *types*	*Tokens* biomédicos	*Types* biomédicos	DeL (%)	DiL-TTR (%)	DiL-VT (%)	Lex-BioMed (%)
Cardiologia	30	37891	8063	9311	912	43,24	21,28	76,28	25,29
Relatório Imagens	9	3717	1414	940	186	40,89	38,04	76,24	24,57
Nefrologia	14	14930	4568	3409	461	42,11	30,60	75,09	22,83
Pediatria	4	5679	2696	1200	390	39,67	47,47	76,31	22,54
Anestesiologia	7	6672	2765	1504	319	40,68	41,44	74,83	21,13
Cirurgia	17	25098	6320	5177	554	42,75	25,18	75,58	20,63
Fonoaudiologia	5	4131	1975	753	203	40,64	47,81	74,27	19,29
Pneumologia	10	4791	2168	924	208	40,94	45,25	74,95	18,23
Terapia Intensiva	11	15534	4147	2563	420	41,53	26,70	75,32	16,50
Outros	16	21446	6103	3485	448	42,41	28,46	75,95	16,25
Epidemiologia	36	50810	8813	6605	448	44,50	17,35	74,24	13,00
Clínica Geral	5	2733	1495	327	100	40,35	54,70	76,11	11,96
Enfermagem	24	46410	9033	5479	416	42,92	19,46	75,84	11,81
Não Classificados	8	6413	2597	660	132	40,78	40,50	75,64	10,29
Saúde Pública	55	83028	14091	7611	534	48,02	16,97	76,20	9,17
Saúde Primária	3	2177	1176	186	61	39,03	54,02	74,30	8,54

Tabela 3. Resultados das características por gênero

Gênero	Total publicações	Total de *tokens*	Total de *types*	*Tokens* biomédicos	*Types* biomédicos	DeL (%)	DiL-TTR (%)	DiL-VT (%)	Lex- BioMed (%)
Aprendendo por Imagem	3	868	484	259	98	45,28	55,76	74,20	29,84
Imagens Pneumologia	35	21521	6846	3652	514	43,90	31,81	74,79	27,35
Relato de Caso	3	8898	3820	972	179	41,88	42,93	76,55	24,85
Artigo de Revisão	14	33093	7350	8121	849	41,80	22,21	74,88	24,54
Recomendações	17	17876	5440	1955	249	42,93	30,43	74,65	22,82
Posicionamento	24	51996	9887	5716	347	42,71	19,01	75,29	22,32
Ponto de Vista	6	7594	3185	911	168	43,29	41,94	76,18	22,14
Cartas ao Editor	8	7607	2773	1684	325	41,38	36,45	75,99	21,13
Revisão	4	4437	2139	419	80	41,60	48,21	76,12	20,16
Nota Técnica	3	1888	992	309	89	44,16	52,54	75,79	19,29
Outros	6	5027	2195	751	124	48,52	43,66	75,50	18,11
Artigo Especial	11	11081	3727	2529	387	39,90	33,63	75,77	17,26
Editorial	20	23653	6410	4284	658	42,67	27,10	76,58	16,97
Comentário	28	11612	4330	2454	448	42,89	37,29	75,30	16,37
Comunicação Breve	6	13652	4125	2752	459	38,49	30,22	73,91	15,95
Artigo de Opinião	6	6854	2873	1703	466	41,39	41,92	73,46	14,94
Reflexão	26	51975	9074	5683	354	41,90	17,46	73,63	14,87
Opinião	8	8556	3037	1365	208	40,51	35,50	75,80	12,00
Artigo	3	479	329	131	62	41,85	68,68	74,11	10,99
Contribuições da Saúde Coletiva	8	9855	3497	1901	322	43,44	35,48	74,00	10,94
Artigo Original	3	6755	2630	1508	360	40,84	38,93	74,88	10,93
Ensaio	9	16900	4936	2917	449	43,17	29,21	75,44	10,92
Perspectiva	3	4863	2132	723	120	41,77	43,84	75,99	9,44

A partir dos resultados da Tabela 3 e da Figura 2(b), pode-se observar também que o léxico de diferentes gêneros pode ser diferenciado com uso da Lex-BioMed, em especial gêneros como “Aprendendo por Imagem” e “Perspectiva”. Na Figura 3 é mostrada a distribuição conjunta entre os gêneros e as especialidades clínicas, onde as linhas e as colunas estão ordenadas por ordem decrescente de Lex-BioMed. Pode-se observar que, *grosso modo*, gêneros com maior valor de Lex-BioMed são constituídos de textos de especialidades que também possuem maior valor do índice, como “Aprendendo por Imagem”, constituído em sua totalidade por textos da especialidade “Relatório Imagens” e “Relato de Caso”, constituído majoritariamente por textos de “Cardiologia”, ambas as especialidades com alta Lex-BioMed. Mesmo gêneros que sugerem textos mais opinativos, como “Posicionamento” e “Ponto de Vista”, possuem relativa alta Lex-BioMed, por serem constituídos exclusivamente por textos de Cardiologia. Além disso, gêneros com baixa Lex-BioMed estão, em geral, associados às especialidades com o mesmo comportamento. Por exemplo, gêneros como “Artigo Original” e “Artigo” possuem maior prevalência de especialidades como “Epidemiologia”, “Enfermagem” e “Saúde Pública”. Os gêneros que possuem os menores valores de Lex-BioMed são “Ensaio” e “Perspectivas”, com textos que descrevem o contexto da pandemia, ao invés de assuntos técnicos do ponto de vista clínico. É o caso

dos textos "COVID-19, as *fake news* e o sono da razão comunicativa gerando monstros: a narrativa dos riscos e os riscos das narrativas" [Vasconcellos-Silva e Castiel, 2020] e "Da Tuberculose ao COVID-19: Legitimidade Jurídico-Constitucional do Isolamento/Tratamento Compulsivo por Doenças Contagiosas em Portugal" [Peixoto et al., 2020]. Os gêneros "Ensaio" e "Perspectivas" possuem maior prevalência de especialidades associadas à baixa Lex-BioMed. A Figura 3 ainda indica uma justificativa para a "Pediatria", especialidade mais geral, possuir valor maior de Lex-BioMed, do que especialidades mais específicas, como, por exemplo, "Pneumologia". Observa-se nesses casos participação majoritária de textos de gêneros como "Relato de Caso", "Artigo de Revisão" e "Imagens Pneumologia", que possuem alta Lex-BioMed como em "Pediatria", enquanto em "Pneumologia" se observa a predominância de textos dos gêneros "Cartas ao Editor" e "Editorial", de menor índice.

Gênero \ Especialidade	Cardiologia (%)	Relatório Imagem (%)	Nefrologia (%)	Pediatria (%)	Anestesiologia	Cirurgia (%)	Fonoaudiologia	Pneumologia (%)	Terapia Intensiva (%)	Outros (%)	Epidemiologia (%)	Clínica Geral (%)	Enfermagem (%)	Não Classificados (%)	Saúde Pública (%)	Saúde Primária (%)
Aprendendo por Imagem		3														
Relato de Caso	4			1			1									
Artigo de Revisão	4			1		5			1		2		1			
Imagens Pneumologia	4			1		5										
Posicionamento	3															
Ponto de Vista	8															
Recomendações			11													
Cartas ao Editor	3				3	2	4	5		4		3		2	1	1
Revisão		1	1							2			1		1	
Outros	5		1	1	2	1		1		2	3		2		2	
Nota Técnica						8										
Editorial	6	1	1	2	2			4	1	3	6	1	3	1	4	
Artigo Especial									3		4			1		1
Comentário									3							
Comunicação Breve	1	1									2				4	
Artigo de Opinião											6					
Reflexão													3			
Opinião											1			1	3	1
Artigo Original						2			3	1	4		12		4	
Artigo										3	5			2	14	
Contribuições da Saúde Coletiva															17	
Ensaio															3	
Perspectivas												1		1	2	

Figura 3: Distribuição conjunta dos textos do corpus por especialidades e gêneros

4. Conclusões

Mostramos neste trabalho que as medidas clássicas de diversidade e densidade lexical não são adequadas para mensurar o léxico de linguagens de especialidade como a biomédica. Com base no conceito de "lexicalidade biomédica", o novo indicador proposto Lex-BioMed foi capaz de revelar a distribuição lexical nos diferentes gêneros e especialidades clínicas presentes no corpus Covid-19 tomado como referência. Os resultados mostraram que os índices de lexicalidade biomédica caem em contextos fronteiriços e mais genéricos em relação a áreas mais técnicas da medicina como a Cardiologia. O recurso de vocabulário teórico utilizado também se revelou interessante para contornar o problema da variação de número de *tokens* entre os textos do corpus.

Dando seguimento à pesquisa, deverá ser buscada uma ampliação do corpus Covid-19 ora composto de 254 textos e restrito à fase inicial de publicações sobre a doença, eventualmente acrescendo-se períodos posteriores que facultem uma análise de teor mais diacrônico. A ampliação do corpus também contempla a possibilidade de estudos comparados com corpora de outra natureza, por exemplo, construídos com base em um léxico mais popular dirigido ao público leigo. Tais resultados sugerem que outras linguagens de especialidade poderão ser investigadas em trabalhos futuros, tendo como horizonte a

hipótese de que seu léxico também seria sensível a uma mensuração de maior aderência ao texto especializado.

Quanto à identificação manual dos termos biomédicos no corpus, cabem algumas considerações finais. Em primeiro lugar, a filtragem dos termos poderá no futuro ser melhor testada mediante consulta ampla a especialistas da área biomédica com experiência em temas associados à Covid-19. Em segundo lugar, a exploração dos termos biomédicos relacionados ao corpus da pandemia poderá ser enriquecida pelo contraponto com vocabulários eletrônicos, glossários e ontologias disponíveis sobre a Covid em português, ou mesmo aqueles em língua estrangeira que possam ser submetidos a tradução. Em terceiro lugar, a caracterização inicial do léxico da Covid baseado apenas em unigramas, tal como apresentado aqui, certamente ganhará se vier a incorporar multipalavras e *collocations* biomédicas. A testagem do indicador Lex-BioMed, de modo a abranger itens nocionais de formação complexa, abrirá caminho para uma abordagem da complexidade vocabular biomédica em associação a outros níveis linguísticos – sintático, semântico, pragmático e discursivo. Por fim, em quarto lugar, um estudo acurado do comportamento das PoS nas várias especialidades e gêneros de que se compõe o corpus estudado poderá refinar a distribuição lexical mostrada aqui. Para além da constatação feita aqui de que os nomes são as "âncoras nocionais" em todas as especialidades clínicas, um estudo de verbos biomédicos como "acometer" ou "diagnosticar", ou de adjetivos como "anestésico" ou "pulmonar", poderá diferenciar classes com regime mais definido ou indefinido de distribuição no corpus.

Agradecimentos

O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES.

Referências

V. Kannan e S. Gurusamy (2014), "Preprocessing Techniques for Text Mining – An Overview", International Journal of Computer Science & Communication Networks, V. 5, p. 7-16.

Leite, J.S., Takahata, A.K., Steinberger-Elias, M.(2020) Elaboração de corpus biomédico em Português sobre o Covid-19. Journal of Health Informatics: Número Especial CBIS Congresso Brasileiro de Informática em Saúde. Dezembro p.242-247. http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/821

Leite, J.S., Takahata, A.K., Steinberger-Elias, M. (2020) "Criação e análise de amostras de corpora em Português Brasileiro para detecção automática de expressões complexas em textos sobre Covid-19". In: XXVII Brazilian Congress on Biomedical Engineering. Proceedings of CBEB 2020, October 26-30, Vitoria, Brazil. https://www.springer.com/gp/book/9783030706005

Orengo, V. & Huyck, C. (2001) "A stemming algorithm for the Portuguese language". In Proceedings of the Eighth International Symposium on String Processing and

Information Retrieval (SPIRE 2001), (p. 186-193). Laguna de San Rafael, Chile: IEEE Computer Society Press.

Aluísio, S. M.; Almeida, G. M. D. B. (2006) "O que é e como se constrói um corpus? Lições aprendidas na compilação de vários corpora para pesquisa linguística Calidoscópio", São Leopoldo, V. 4, n.3. p.155-177.

Celso Romão Cardoso De Almeida Júnior (2017). "Proposta de um Sistema Automático de Avaliação de Redações do Enem, Foco na Competência 1: Demonstrar Domínio da Modalidade Escrita Formal da Língua Portuguesa". Dissertação de Mestrado.

Cucinotta, Domenico, and Maurizio Vanelli. (2020) "WHO declares COVID-19 a pandemic". Acta Bio Médica: Atenei Parmensis 91.1 (2020): p.157.

Vasconcellos-Silva, P. R., & Castiel, L. D. (2020) "COVID-19, "As fake news e o sono da razão comunicativa gerando monstros: a narrativa dos riscos e os riscos das narrativas". Cadernos de Saúde Pública, V. 36, n. 7, p.1-6.

Peixoto, V. R., Mexia, R., Santos, N. D. S., Carvalho, C., & Abrantes, A. (2020) "Da tuberculose ao COVID-19: legitimidade jurídico-constitucional do isolamento/tratamento compulsivo por doenças contagiosas". In Portugal. Acta Médica Portuguesa, V. 33, p.225-228.

Krieger, M. da G, Finatto, M. J. B. (2004) "Introdução à terminologia: teoria & prática". São Paulo:Contexto, p.348.

Ure, J. (1971) "Lexical density and register differentiation". In: G.E. PERREN; J.L.M. TRIM (eds.), Applications of linguistics. Selected papers of the Second International Congress of Applied Linguistics. Cambridge/Londres, Cambridge University Press, p. 443-452.

Johansson, V. (2008) "Lexical diversity and lexical density in speech and writing: a developmental perspective". Lund University, Department of Linguistics and Phonetics: Working Papers, V. 53, p.61-79.

Broeder, P., Coenen, J., Extra, G., van Hout, R., & Zerrouk, R. (1986) "Ontwikkelingen in het Nederlandstalig lexicon bij anderstalige volwassenen: Een macro- en microperspectief". In J. Creten, G. Geerts, & K. Jaspaert (Eds.), Werk-in-uitvoering: Momentopname van de sociolinguïstiek in België en Nederland, p.39-57.

Perna, L. Cristina; Delgado, K. Heloísa; Finatto, J. Maria. (2010) "Linguagens Especializadas em CORPORA. Modos de Dizer e Interfaces de Pesquisa". EDIPUCS-Editora Universitária da Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, p.138.

Santos, E. S. et al. (2018) "Diversidade e densidade lexical em textos escritos por alunos recém-alfabetizados: um estudo descritivo de produções individuais e em díades". Calidoscópio Revista Unisinos, V. 16, n.1, p.25-32.

Silva W.D.C.M. (2013) "Aprimorando o corretor gramatical CoGrOO", Dissertação de Mestrado em Ciência da Computação, IME-USP, São Paulo, SP.

Zilio, L. (2009) "Colocações especializadas e Komposita: um estudo contrastivo alemão-português na área de cardiologia". Porto Alegre: UFRGS. Dissertação de Mestrado. PPG-LETRAS/UFRGS.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Análise de polaridade e de tópicos em *tweets* no domínio da política no Brasil

Leonardo Capellaro[1], **Helena de Medeiros Caseli**[1]

[1]Universidade Federal de São Carlos (UFSCar)
Departamento de Computação – LALIC
Caixa Postal 676 – 13565-905 – São Carlos – SP – Brasil

leonardo.capellaro@gmail.com, helenacaseli@ufscar.br

Abstract. *The field of politics in Brazil is one of the busiest and most controversial in the last decade. With the advent of social networks, a new communication channel between voters and politicians was created, with users having a space to publish their opinions and beliefs. In this context, this work shows the performance of BERT models in the polarity (positive, negative or neutral) and topic analysis of tweets related to Brazilian politics. Experiments were caried out with tweets related to the 2018 elections achieving results ($F1 = 96\%$) better than the ones obtained by previous works for the polarity classification. The qualitative analysis of the topics was also promising.*

Resumo. *O campo da política no Brasil é um dos mais movimentados e mais polêmicos da última década. Com o advento das redes sociais, um novo canal de comunicação entre os eleitores e os políticos foi criado, com os usuários tendo um espaço para publicar suas opiniões e crenças. Neste contexto, este trabalho mostra como modelos BERT se saem na análise de polaridade (positiva, negativa ou neutra) e de tópicos de grandes quantidades de tweets relacionados à política do Brasil. Experimentos foram realizados com tweets relacionados às eleições de 2018, e melhores resultados ($F1 = 96\%$) foram obtidos para a classificação de polaridade em comparação com trabalhos anteriores. A avaliação qualitativa dos tópicos também mostrou resultados promissores.*

1. Introdução

Um estudo realizado pelo IBGE[1], em 2019, demonstrou que 82,7% dos domicílios brasileiros possuía conexão com a internet. Outro estudo realizado no mesmo ano pela Comscore[2] apontou que aproximadamente 88% da população brasileira tinha acesso a algum tipo de rede social, com o Twitter possuindo 40,2 milhões de usuários ativos no país. Por ser uma rede social que limita o número de caracteres nos textos postados na rede, o Twitter se tornou um canal de comunicação rápida e simplificada, sendo uma boa ferramenta para saber o que as pessoas estão comentando espontaneamente sobre os assuntos do momento [Sales e Barbosa 2019].

[1]https://educa.ibge.gov.br/jovens/materias-especiais/20787-uso-de-internet-televisao-e-celular-no-brasil.html

[2]https://olhardigital.com.br/2019/07/05/noticias/brasil-e-o-pais-que-mais-usa-redes-sociais-na-america-latina/

No domínio da política, as redes sociais também passaram a ser uma importante vitrine para os candidatos divulgarem seus trabalhos, suas propostas e opiniões e atingirem mais rapidamente seus eleitores. De acordo com um estudo divulgado pelo próprio Twitter em 2018[3], no período de 16 de agosto a 28 de outubro de 2018, foram contabilizados 165 milhões de *tweets* relacionados às eleições, um volume quatro vezes maior que o total das eleições anteriores, em 2014.

Saber "o que sentem" e "sobre o que falam" os usuários do Twitter, no domínio da política brasileira, é a grande motivação deste trabalho. Para tanto, este trabalho apresenta resultados da investigação de análise de polaridade e de tópicos em *tweets*, escritos em português, no domínio da política brasileira coletados em 2018. A **análise de polaridade** consiste em determinar qual a polaridade (ou valência) da opinião do autor de um texto com relação à entidade ou assunto em discussão. Desse modo é possível, por exemplo, verificar se a opinião do autor é positiva, negativa ou neutra em relação à entidade ou assunto alvo do texto. A **análise de tópicos**, por sua vez, visa lidar com grandes quantidades de textos, análises e *feedbacks*, de forma a retornar os principais assuntos/tópicos dos textos de acordo com sua importância e recorrência.

Alguns trabalhos anteriores foram realizados analisando sentimentos em *tweets* do campo da política, como [Moreira et al. 2020] e [Cristiani et al. 2020]. Em [Moreira et al. 2020], foi proposta uma análise da polarização da elite em comparação com a massa no procedimento de impeachment da presidente do Brasil em 2016. Em [Cristiani et al. 2020], a análise de sentimentos é aplicada a *tweets* das eleições presidenciais de 2018 no Brasil, onde foi estudada a relação entre as opiniões dos usuários que publicaram sobre os candidatos durante o período das eleições e seu resultado final. Entretanto, devido à ausência de um modelo de linguagem contextualizado (considerado o estado-da-arte para diversas aplicações de PLN) treinado/disponível para o português do Brasil naquela época, como o BERTimbau [Souza et al. 2020], outros métodos foram utilizados, como o SVM e Naive Bayes.

Assim, as principais contribuições deste trabalho estão relacionadas à análise de desempenho de modelos BERT no *corpus* de [Cristiani et al. 2020] para: (i) a análise de polaridade usando o BERTimbau [Souza et al. 2020], dimensionando o avanço em performance com o uso desta nova técnica; e (ii) a análise de tópicos utilizando o BERTopic [Grootendorst 2020], algo ainda inédito neste *corpus* e domínio.

2. Trabalhos relacionados

Esta seção traz uma visão geral de alguns trabalhos realizados para análise de sentimentos e de tópicos em *tweets*, escritos em português do Brasil, no domínio da política.

2.1. Análise de sentimentos

Podemos encontrar diversos trabalhos que realizaram a análise de sentimentos em *tweets* na língua portuguesa. Em [Christhie et al. 2018], os autores utilizaram Naive Bayes, SVM e Random Forest para realizar uma análise de posicionamento em *tweets* de diversos candidatos às eleições de 2018, classificando-os em: contra, a favor ou neutro. O algoritmo SVM conseguiu uma $F1$ de 99% em um dos conjuntos utilizados na análise.

[3] https://blog.twitter.com/pt_br/topics/company/2018/como-foram-as-eleicoes-2018-no-twitter

Em [Pereira 2019], o autor realizou uma análise dos sentimentos dos *tweets* para cada um dos candidatos das eleições de 2018, separados por evento, comparando o sentimento em todos os debates realizados no período e analisando a popularidade de cada candidato com base nos sentimentos positivos dos textos da rede social. Foram utilizados os métodos SVM e Naive Bayes, onde o método SVM atingiu 86% de acurácia no melhor caso, enquanto o Naive Bayes ficou em 85%.

Já em [Cristiani et al. 2020], os autores utilizaram Naive Bayes e SVM em um *corpus* com aproximadamente 369 mil *tweets* sobre as eleições presidenciais de 2018. O trabalho buscava relacionar os sentimentos identificados nos *tweets* a respeito de cada candidato com o resultado final das eleições, e os valores de $F1$ para os métodos Naive Bayes e SVM ficaram em $54,2\%$ e $66,1\%$ respectivamente.

No presente trabalho, a análise de polaridade dos *tweets* do *corpus* de [Cristiani et al. 2020] foi realizada com o *fine-tuning* de um modelo BERT pré-treinado em português do Brasil, o BERTimbau [Souza et al. 2020], lançando-se mão de uma série de técnicas de pré-processamento com o intuito de obter a melhor $F1$ para o modelo. O modelo foi treinado com 600 *tweets* anotados manualmente, e posteriormente aplicado em um *corpus* contendo aproximadamente 370 mil *tweets* para classificá-los.

2.2. Análise de tópicos

Diversas abordagens para a análise de tópicos foram empregadas em trabalhos relacionados. Em [Pinto et al. 2020], os autores utilizaram modelagem de tópicos e análise de sentimentos em *tweets* da língua inglesa durante a pandemia da COVID-19. Os pesquisadores utilizaram três algoritmos de aprendizagem de máquina: o *Latent Dirichlet Allocation* (LDA), o *Non-Negative Matrix Factorization* (NMF) e o *Latent Semantic Analysis* (LSA). Foi realizada uma análise qualitativa dos resultados, onde o NMF foi o algoritmo que gerou a melhor relação entre os termos e a maior coerência com o tema da pesquisa.

O BERTopic [Grootendorst 2020] foi utilizado em [Silveira et al. 2021] para fazer uma análise de tópicos em documentos do meio jurídico. Na avaliação qualitativa, realizada por especialistas, constatou-se que $84,6\%$ dos tópicos gerados pelo modelo correspondiam aos temas principais dos documentos.

No presente trabalho, foi utilizado o BERTopic para realizar a modelagem de tópicos com o intuito de obter os principais assuntos mencionados nos *tweets* previamente anotados automaticamente como positivos e negativos no domínio da política.

3. Metodologia

Esta seção descreve os métodos empregados para análise de polaridade e de tópicos em *tweets* do domínio da política, escritos em português do Brasil.

3.1. Análise de polaridade com BERT

O BERT (*Bidirectional Encoder Representations from Transformers*) é um algoritmo que emprega treinamentos bidirecionais de *Transformers*, e foi proposto em [Devlin et al. 2019]. O algoritmo possui como característica fundamental o fato de ser bidirecional, o que permite uma maior compreensão do contexto de uma palavra e do texto como um todo, analisando as palavras adjacentes em ambas as direções.

Na maioria das aplicações práticas são utilizados modelos pré-treinados do BERT, que podem ser treinados em uma única língua ou em diversas línguas diferentes. O modelo pré-treinado passa, então, por uma etapa de ajuste ou sintonia fina (*fine-tuning*), que modela a última camada da rede neural BERT para a especificidade do problema em questão, como ilustrado na Figura 1. Desta forma, o modelo pré-treinado é criado utilizando grandes *corpora* e máquinas com grandes capacidades de processamento, deixando para o usuário apenas a necessidade de realizar o *fine-tuning* para um problema específico.

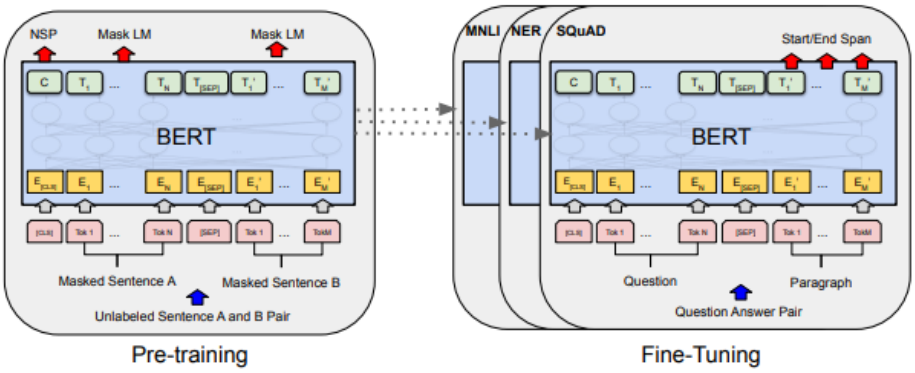


Figura 1. Processo de pré-treinamento e *fine-tuning* do BERT, de acordo com [Devlin et al. 2019]

A análise de polaridade de textos é tratada como uma tarefa de classificação, onde cada texto possui um rótulo indicando o tipo de sentimento empregado na escrita. Nesse tipo de tarefa, o BERT funciona adicionando uma camada de classificação na parte superior da saída do *Transformer*. A maioria dos hiperparâmetros da rede neural se mantém os mesmos do modelo pré-treinado previamente, com exceção do tamanho de lote (*batch*), da taxa de aprendizagem (*learning rate*) e do número de períodos de treinamento (*epochs*).

Para o presente trabalho, foi utilizado o BERTimbau [Souza et al. 2020], modelo pré-treinado para o português do Brasil e disponibilizado pela neuralmind[4]. Para o *fine-tuning* do modelo, foi utilizado o *corpus* disponibilizado por [Cristiani et al. 2020][5].

3.2. Análise de tópicos usando BERT e TF-IDF

Para a análise de tópicos, foi utilizado o algoritmo BERTopic[6] [Grootendorst 2020]. Este algoritmo utiliza as *embeddings* de palavras presentes em um modelo BERT para gerar agrupamentos hierárquicos por densidade de palavras utilizando o HDBSCAN e, por fim, gera tópicos com base na importância das palavras utilizando uma variação do TF-IDF, chamada de c-TF-IDF.

A primeira etapa do BERTopic converte os documentos em dados numéricos com base nas *embeddings* do modelo BERT. A segunda etapa é a de clusterização, que é realizada através do HDBSCAN, um algoritmo de agrupamento hierárquico por densidade

[4]https://github.com/neuralmind-ai/portuguese-bert
[5]https://github.com/andrecristiani/analise-de-sentimentos-eleicoes-2018
[6]https://maartengr.github.io/BERTopic/index.html

proposto em [Campello et al. 2013]. Neste algoritmo, os documentos que possuem maior similaridade entre si são agrupados em *clusters* baseados na estabilidade do *cluster*. Uma das importantes características do HDBSCAN é o fato de ele não forçar a seleção de um dado para um determinado *cluster*. Caso o dado não se encaixe em nenhum grupo por similaridade, ele é considerado um *outlier*.

A última etapa do processo é a seleção dos tópicos com base na importância das palavras. Para isso, o autor desenvolveu uma técnica nomeada de *c-TF-IDF*. Esta técnica funciona de forma muito parecida com o *TF-IDF* original, que compara a importância das palavras analisando todo o *corpus*, entretanto, o *c-TF-IDF* utiliza os *clusters* gerados na etapa de agrupamento aplicando o *TF-IDF* em cada um deles. Esse processo classifica as palavras de acordo com a importância para cada grupo gerado no processo de agrupamento, gerando os principais tópicos de cada grupo.

4. Experimentos

Esta seção descreve as configurações dos experimentos, com a descrição do *corpus* e suas etapas de pré-processamento, e as questões de pesquisa.

4.1. *Corpus*

O *corpus* de treinamento utilizado neste trabalho é o mesmo de [Cristiani et al. 2020]. Esse *corpus* está composto por 600 *tweets* anotadas manualmente com suas respectivas polaridades: positivo, quando a mensagem publicada demonstra apoio ao candidato; negativo, quando a mensagem publicada demonstra rejeição ao candidato; e neutro, quando a mensagem publicada não demonstra opinião de polaridade clara sobre o candidato.

Em [Cristiani et al. 2020], todas as letras dos textos do *corpus* foram transformadas em minúsculas, todos os *retweets* dos usuários foram removidos e qualquer tipo de *hiperlink* incluído na mensagem foi apagado. No presente trabalho, utilizou-se a ferramenta Enelvo[7] [Bertaglia e Nunes 2016], que visa a padronização (normalização) dos textos obtidos da Internet. Em testes preliminares, a configuração de pré-processamento que apresentou melhores resultados foi a que denominamos "Enelvo Raw sem emoji e retweet", onde os textos foram normalizados pelo Enelvo com o parâmetro *tokenizer* configurado como *readable*, mantendo as entidades dos textos inalteradas[8] e, em seguida, foram removidos emojis e *retweets*. Assim, nos experimentos apresentados neste artigo, dois pré-processamentos foram comparados:

- Original: pré-processamento utilizado em [Cristiani et al. 2020], onde são removidos *hiperlinks*, *retweets* e o texto é convertido para minúsculas.
- Enelvo Raw sem emoji e retweet: normalização feita no modo Enelvo Raw, mas removendo emojis e *retweets*.

A Tabela 1 traz exemplos de *tweets* originais do *corpus* de [Cristiani et al. 2020].

Além do *corpus* com 600 *tweets* anotados com polaridade, neste trabalho também foi utilizado o *corpus* de 369 mil *tweets* de [Cristiani et al. 2020], sem anotação de polaridade, que foram obtidos durante o segundo turno das eleições. Esse *corpus* foi usado para

[7] https://thalesbertaglia.com/enelvo/

[8] Por padrão, o Enelvo troca algumas entidades do texto por *tags*, como o nome do usuário por "username", *hashtags* por "hashtag" e números por "number".

Tabela 1. Exemplos de *tweets* anotados manualmente com polaridade

Texto do *tweet*	**Classe**
a justiça deveria ser sempre ágil, como é para tudo relacionado ao lula. só acho.	neutro
fui de #ciro no primeiro turno. parte não é minha preferência. mas agora sou #haddadpresidente desde criancinha. #eleições2018	positivo
realmente bolsonaro não tem raciocínio lógico. não conseguiu sair da primeira pergunta. #bolsonaronojornalnacional	negativo

Fonte: *Tweets* extraídos do *coprus* de [Cristiani et al. 2020]

a análise de tópicos relativos aos dois candidatos do segundo turno. Devido ao grande tamanho do *corpus*, não foi possível processá-lo com o Enelvo.[9] Assim, um processamento mais simples foi realizado para: (i) remover *hashtags* e links através de expressões regulares; (ii) converter para minúsculas; (iii) remover os inícios de parágrafos representados por \n; (iv) remover vogais repetidas nas palavras (por exemplo, 'corruptooo' foi transformado em 'corrupto'); (v) remover as pontuações do texto (vírgula, interrogação, exclamação, pontos finais, dentre outras); (vi) remover as *stopwords* (como "o", "a", "com", "para", entre outras); e (vii) remover as palavras com apenas uma letra.

4.2. Configurações dos experimentos

Para a classificação de polaridade dos *tweets*, os algoritmos de classificação selecionados foram: BERT, Naive Bayes e SVM, sendo os dois últimos utilizados em [Cristiani et al. 2020]. Os *corpora* utilizados foram o original de [Cristiani et al. 2020] e o pré-processado Enelvo Raw sem emoji e *retweet*.

No treinamento, o *corpus* foi dividido em: 90% para treinamento e 10% para teste, utilizando a mesma partição dos dados para todos os algoritmos investigados. O BERT foi refinado por 15 épocas, sendo considerado o melhor valor de $F1$ entre as épocas para a comparação com os demais algoritmos. A taxa de aprendizagem para o BERT foi fixada em 2e-5 e foi utilizado o otimizador AdamW. Buscando replicar os mesmos parâmetros utilizados por [Cristiani et al. 2020], para a classificação com o SVM foi utilizado o kernel linear, parâmetro de regularização C igual a 1, grau da função polinomial do kernel em 3 e valor de gamma em "auto". Para o Naive Bayes foram utilizados os parâmetros *default* do modelo multinomial do `scikit-learn`[10].

Na análise de tópicos, utilizou-se o BERTopic com o modelo BERTimbau limitando o número de tópicos a 20, quantidade que foi definida empiricamente analisando os resultados gerados pelo modelo completo, que mostrou ter aproximadamente este número de tópicos. Por ser um método de classificação não supervisionado, não precisa de nenhum treinamento e utiliza apenas dados não rotulados.

5. Resultados

Os experimentos foram divididos em três etapas para responder as seguintes questões de pesquisa:

Q1 O desempenho do BERT na análise de polaridade supera o de Naive Bayes e SVM utilizando o mesmo *corpus* e pré-processamento de [Cristiani et al. 2020]?

[9] O Enelvo ficou em execução durante 4 dias e não concluiu o processamento do *corpus*.
[10] `https://scikit-learn.org/`

Q2 O pré-processamento refinado, com o auxílio da ferramenta Enelvo, traz ganho de desempenho de BERT, Naive Bayes e SVM na análise de polaridade?

Q3 O BERTopic pode ser considerado uma boa ferramenta para a extração de tópicos, em *tweets* classificados como positivos e negativos referentes a cada candidato, com base em uma análise qualitativa dos dados?

5.1. Análise de polaridade

A Tabela 2 resume os resultados dos experimentos realizados com a análise de polaridade, para responder as questões de pesquisa Q1 e Q2.

Tabela 2. Resultados da avaliação de análise de polaridade

Algoritmo	***Corpus* pré-processado**	$F1$	
Naive Bayes	Original	47,5%	Q1
Naive Bayes	Enelvo Raw sem emoji e *retweet*	53,1%	Q2
SVM	Original	55,5%	Q1
SVM	Enelvo Raw sem emoji e *retweet*	57,8%	Q2
BERT	Original	92,4%	Q1
BERT	Enelvo Raw sem emoji e *retweet*	96,6%	Q2

De acordo com os valores de $F1$ apresentados nesta tabela, é possível notar que o BERT obteve um resultado superior comparado aos algoritmos até então investigados para esse *corpus* em [Cristiani et al. 2020]. BERT alcançou uma $F1$ de 96,6% enquanto o melhor desempenho relatado em [Cristiani et al. 2020] tinha sido de 66,2%, utilizando SVM. No experimento deste trabalho, o algoritmo SVM conseguiu uma F1 de 57,8%. Assim, esses resultados respondem a primeira questão de pesquisa **Q1** mostrando que o BERT, refinado para o problema de análise de polaridade com o *corpus* de [Cristiani et al. 2020], apresenta o novo melhor desempenho obtido neste *corpus*.[11]

Quanto à **Q2**, nota-se que os resultados de $F1$ obtidos para os diferentes processamentos do *corpus* ficaram próximos para os diferentes modelos. Contudo, em todos os casos, o *corpus* Enelvo Raw sem emoji e *retweet* se saiu um pouco melhor, especialmente no Naive Bayes. Assim, respondendo à **Q2**, conclui-se que o pré-processamento proposto e adotado neste trabalho traz ganhos para a tarefa de análise de polaridade.

5.2. Análise qualitativa de tópicos

Para a análise qualitativa de tópicos usando o BERTopic, o *corpus* com 369.800 *tweets* descrito na seção 4.1 foi processado pelo modelo de análise de polaridade treinado com o BERT. Dos 369.800 *tweets*, 346.243 obtiveram rótulos gerados pelo modelo (um aproveitamento de $93,6\%$). Destes, $54,7\%$ são de *tweets* citando o candidato Jair Bolsonaro e $45,3\%$ citando o candidato Fernando Haddad. Cada um dos *tweets* foi classificado em positivo, negativo, neutro ou em mais de um rótulo.

[11]Os valores de SVM e Naive Bayes apresentados na Tabela 2 diferem dos apresentados em [Cristiani et al. 2020] porque não foi possível replicar a partição de treino e teste utilizada no experimento original. Entretanto, mesmo tendo resultados um pouco abaixo nos métodos SVM e Naive Bayes na experimentação deste trabalho, é possível afirmar que o uso do BERT gerou um salto significante de $F1$, indo para os 96,6%, enquanto a $F1$ máxima obtida pelo método SVM no trabalho de [Cristiani et al. 2020] foi de 66,1%.

A quantidade de *tweets* rotulados como positivos ($59,4\%$) é maior do que a de $tweets$ negativos ($21,6\%$) e neutros ($18,1\%$).[12] O candidato Fernando Haddad ficou com uma maior quantidade dos *tweets* positivos ($31,3\%$ contra $28,1\%$) e uma menor quantidade de *tweets* negativos ($7,7\%$ contra $13,9\%$) comparado ao candidato Jair Bolsonaro.

O *corpus* anotado automaticamente com polaridade foi, então, dividido em quatro conjuntos distintos de *tweets* para realizar a análise de tópicos: `Bolsonaro+`, `Bolsonaro-`, `Haddad+` e `Haddad-`. Para cada um desses conjuntos foram gerados 20 tópicos com 5 palavras cada. Destes 20 tópicos, foram selecionados os 9 primeiros, numerados de 0 a 8, uma vez que quanto maior o número do tópico, menos frequente ele é. As palavras dentro de cada tópico são ordenadas pela relevância gerada pelo *c-TF-IDF*. A Tabela 3 resume os principais resultados da análise de tópicos nestes quatro conjuntos.

Tabela 3. Análise de tópicos extraídos pelo BERTopic

Conjunto	Tópicos	Análise
`Bolsonaro+`	carvalho, hasselmann, paschoal	Partes dos nomes de alguns dos principais aliados de Bolsonaro em 2018: Olavo de Carvalho, Joyce Hasselmann e Janaína Paschoal
	acabouapiranhagempt, vitória, imparcial	Partes de *slogans* ou frases feitas de campanha, além de características mencionadas por possíveis apoiadores de Bolsonaro
`Bolsonaro-`	homofóbicos, fascistas, desrespeitarem, ódio	Termos encontrados em *tweets* com polaridade (argumentos) negativos em relação a Bolsonaro
`Haddad+`	haddadpresidente, trabalhando, vença, democracia, respeito, imparcial	Partes de *slogans* ou frases feitas de campanha, além de características mencionadas por possíveis apoiadores de Haddad
	indecisos	A palavra "indecisos" também apareceu, uma vez que naquela época uma das chances de virada de Haddad era na conversão dos votos de pessoas indecisas
`Haddad-`	xingando, meme	Poucas palavras que levam à conotação negativa apareceram relacionadas a Haddad, enquanto o restante foram palavras neutras, como "campanha", "votando" e "falando".

Sobre a **Q3**, pode-se concluir que o BERTopic se mostrou uma ferramenta bastante promissora para a extração de tópicos nos diferentes conjuntos, mostrando tópicos que faziam sentido em todos os cenários. Porém, vale ressaltar que a análise de tópicos foi a ponta final do experimento, carregando erros tanto de associação do *tweet* ao candidato correto quanto de falsos positivos e falsos negativos na classificação pelo modelo BERT.

6. Trabalhos futuros

Considerando os trabalhos futuros, há uma margem para melhorias nos métodos de pré-processamento, visto que o uso do Enelvo se tornou computacionalmente inviável para grandes quantidades de texto. Pode-se estudar métodos de melhoria de desempenho deste algoritmo utilizando técnicas de paralelização, por exemplo. Também existe uma margem para melhorias na extração de tópicos, buscando métodos de identificação de entidades e removendo as entidades neutras ou com baixo significado. Vale ressaltar que a metodologia e as ferramentas empregadas neste trabalho podem ser utilizadas em outros tipos de pesquisa relacionadas a conteúdos textuais em língua portuguesa.

[12] $0,9\%$ dos *tweets* foram rotulados em duas classes.

Referências

Bertaglia, T. F. C. e Nunes, M. d. G. V. (2016). Exploring word embeddings for unsupervised textual user-generated content normalization. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 112–120.

Campello, R. J. G. B., Moulavi, D., e Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In Pei, J., Tseng, V. S., Cao, L., Motoda, H., e Xu, G., editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg. Springer Berlin Heidelberg.

Christhie, W., Reis, J. C. S., Moro, F. B. M. M., e Almeida, V. (2018). Detecção de posicionamento em tweets sobre política no contexto brasileiro. In *Anais do VII Brazilian Workshop on Social Network Analysis and Mining*, Porto Alegre, RS, Brasil. SBC.

Cristiani, A., Lieira, D., e Camargo, H. (2020). A sentiment analysis of brazilian elections tweets. In *Anais do VIII Symposium on Knowledge Discovery, Mining and Learning*, pages 153–160, Porto Alegre, RS, Brasil. SBC.

Devlin, J., Chang, M.-W., Lee, K., e Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*, page 4171–4186.

Grootendorst, M. (2020). BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics.

Moreira, R., Vaz de Melo, P., e Pappa, G. (2020). Elite versus mass polarization on the brazilian impeachment proceedings of 2016. *Social Network Analysis and Mining*, 10(92).

Pereira, J. G. (2019). Análise de sentimentos da população brasileira em relação a eleição presidencial de 2018 através da rede social twitter. Trabalho de Conclusão de Curso (Bacharelado) - Universidade Federal do Rio Grande do Norte. Centro de Ensino Superior do Seridó. Departamento de Computação e Tecnologia.

Pinto, M. A. S., Junior, A. F. L. J., Busson, A. J. G., e Colcher, S. (2020). Relacionando modelagem de tópicos e classificação de sentimentos para análise de mensagens do twitter durante a pandemia da covid-19. In *Anais Estendidos do XXVI Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 61–64, Porto Alegre, RS, Brasil. SBC.

Sales, M. L. e Barbosa, M. W. (2019). Uma avaliação do potencial de uso dos dados do Twitter para a predição do resultado de eleições: O caso das eleições presidenciais brasileiras de 2018. *Revista de Informática Aplicada - RIA*, 15(2):30–43.

Silveira, R., Fernandes, C. G., Neto, J. A. M., Furtado, V., e Filho, J. E. P. (2021). Topic modelling of legal documents via legal-bert1. In *RELATED - Relations in the Legal Domain Workshop, in conjunction with ICAIL 2021*, São Paulo, Brazil.

Souza, F., Nogueira, R., e Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *Lecture Notes in Computer Science – Proceedings of the 9th Brazilian Conference on Intelligent Systems (BRACIS)*, volume 12319, pages 403–417.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Utilizando BERTimbau para a Classificação de Emoções em Português

Luiz Otávio Alves Hammes[1], **Larissa Astrogildo de Freitas**[1]

[1]Centro de Desenvolvimento Tecnológico (CDTec)
Universidade Federal de Pelotas (UFPel) – Pelotas, RS – Brasil

{loahammes,larissa}@inf.ufpel.edu.br

Abstract. *In this work, we propose the fine-tuning of the BERTimbau-base and BERTimbau-large models in the task of Sentence Classification with 27 types of emotions, based on the GoEmotions dataset translated into Portuguese, using automatic translation tools. We compared the results of our experiments with the one provided by the authors of the GoEmotions dataset and obtained a performance gain which we attribute to the balancing algorithm used.*

Resumo. *Neste trabalho propomos realizar o fine-tuning dos modelos BERTimbau-base e BERTimbau-large na tarefa de Classificação de 27 tipos de emoções em sentenças, baseado no dataset GoEmotions traduzido para a língua portuguesa, por meio de ferramentas de tradução automática. Comparamos os resultados de nossos experimentos com os resultados disponibilizados pelos autores do dataset GoEmotions e obtivemos um ganho de desempenho ao qual atribuímos ao algoritmo de balanceamento utilizado.*

1. Introdução

Análise de Sentimento (AS) é uma tarefa da área de Processamento de Língua Natural (PLN), que tem como objetivo extrair, processar e classificar opiniões, sentimentos, emoções e avaliações da vasta quantidade de conteúdo disponível na Web atualmente, podendo ser textos, áudios ou imagens [Liu 2012].

Ainda, podemos dividir as tarefas de AS em três níveis: (1) nível de documento, no qual um sentimento é atribuído ao documento como um todo ; (2) nível de sentença, no qual um sentimento é atribuído a cada sentença presente em um documento; (3) nível de aspecto, no qual são identificados aspectos de uma entidade[1] presentes em um documento e então um sentimento é atribuído para cada um dos aspectos encontrados.

A Classificação de Emoções (CE) em nível de sentença é uma subárea da AS, a qual tem como objetivo classificar as distintas emoções expressas em um conjunto de sentenças que, em sua grande maioria, são de caráter subjetivo. Isso acontece devido as sentenças subjetivas manifestarem emoções, que estão atreladas as percepções, os pontos de vista e o estado emocional dos seus autores [Liu 2012].

Definir o conceito de emoção é uma tarefa difícil. Isso ocorre devido as emoções estarem relacionadas a uma imensa quantidade de distintos fatores que podem fazer com que elas sejam manifestadas. Uma das principais teorias é a psicoevolutiva, a qual sugere

[1]Uma entidade é algo que pode ser nomeado, como: pessoa, objeto, produto, localização, etc.

a existência de manifestações básicas de emoções que são consequência de processos evolutivos, os trabalhos de [Ekman 2004] e de [Plutchik 2003] defendem essa ideia.

Avanços recentes na psicologia sugerem a existência de um conjunto maior de emoções, manifestadas a partir de diferentes contextos e situações. Por exemplo, o trabalho de [Cowen and Keltner 2017] identificou 27 emoções expressas por pessoas a partir de vídeos curtos e o trabalho de [Cowen and Keltner 2020] identificou 28 emoções expressas através de expressões faciais e linguagem corporal.

Tendo em vista este contexto, percebemos que é necessário ampliar a quantidade de classes de emoções dos modelos classificadores disponíveis na literatura para abranger estes estudos recentes, já que a maioria dos trabalhos utiliza como base apenas os trabalhos de [Ekman 1992] (6 emoções básicas) e de [Plutchik 1982] (8 emoções básicas).

Neste trabalho, propomos utilizar modelos de Aprendizado de Máquina (AM) para classificar 27 emoções com base no *dataset* GoEmotions [Demszky et al. 2020], traduzido para o português, com a finalidade de melhorar os resultados preliminares e ampliar a variedade de classes na tarefa de CE em língua portuguesa, disponíveis na literatura.

Este artigo está estruturado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 descreve a metodologia; a Seção 4 mostra a análise dos resultados; e a Seção 5 apresenta as conclusões.

2. Trabalhos Relacionados

De acordo com [Pereira 2021], poucos são os trabalhos encontrados na literatura que focam na tarefa de CE em língua portuguesa. Dentre eles podemos citar o trabalho de [Duarte et al. 2019] e de [Dosciatti et al. 2013].

No trabalho de [Duarte et al. 2019] foram utilizados emojis para reconhecer 6 emoções básicas (felicidade, raiva, nojo, medo, tristeza e surpresa) utilizando a taxonomia de [Ekman 1992] em *tweets*. Os autores coletaram 2 milhões de *tweets* que continham pelo menos um dos emojis que segundo [Wood and Ruder 2016] são indicadores da presença das 6 emoções básicas. Em seguida realizaram um etapa de pré-processamento, na qual foram retirados *retweets, URLs*, menções de usuários, *tweets* com dois ou mais emojis que expressam emoções contraditórias e *tweets* com menos de 3 *tokens*. Os emojis foram retirados dos *tweets* e utilizados como *labels* para treinamento dos modelos classificadores. Foram treinados dois modelos: *Support Vector Machine* (SVM) e *Naive Bayes* (NB), atingindo 0,62 e 0,70 de Medida-F, respectivamente.

No trabalho de [Dosciatti et al. 2013] foi usado um corpus com 1750 notícias curtas para treinar um modelo classificador, rotuladas nas 6 emoções básicas de [Ekman 1992] mais a classe "neutro"[2]. Inicialmente, os autores coletaram notícias do site `www.globo.com`, depois realizaram uma etapa de pré-processamento, na qual retiraram acentos, caracteres especiais e *stopwords* e converteram todos os textos para letras minúsculas. Em seguida, dois anotadores classificaram os textos nas 6 emoções básicas ou na classe "neutro". O corpus foi construído mantendo uma proporção entre as 7 classes para evitar desbalanceamento, resultando em 1750 notícias, 250 para cada classe. Os autores treinaram o modelo SVM, atingindo 0,60 de Medida-F.

[2]A classe "neutro" representa um texto que não contém conteúdo emocional.

No contexto da língua inglesa, podemos citar o trabalho de [Demszky et al. 2020] que desenvolveu o *dataset* GoEmotions com taxonomia de 27 emoções, baseando-se nos trabalhos de [Cowen and Keltner 2017], [Cowen et al. 2019a], [Cowen and Keltner 2020], [Cowen et al. 2019b], contendo 54263 sentenças. Ainda, os autores disponibilizaram um modelo BERT-base [Devlin et al. 2018] com *fine-tuning* utilizando o *dataset* GoEmotions. Mais detalhes sobre o *dataset* GoEmotions serão discutidos na Seção 3, já que ele será utilizado como base para nosso trabalho.

3. Metodologia

O processo de execução desse trabalho foi divido em três etapas principais. A concepção do modelo, *fine-tuning* e avaliação dos resultados obtidos.

Em nossos experimentos utilizamos a versão filtrada do *dataset* GoEmotions, que contém 54263 sentenças manualmente anotadas em 28 classes (27 emoções + neutro) retiradas do fórum Reddit na língua inglesa. O processo de filtragem aplicado por [Demszky et al. 2020] é realizado em duas etapas, primeiro são retirados todos os *labels* que foram selecionados por apenas um anotador e depois são mantidas apenas as sentenças com pelo menos um *label*. Manteve-se a proporção original: 80% treinamento, 10% validação e 10% teste.

Como o objetivo deste trabalho é CE em sentenças escritas em português, traduzimos o *dataset* GoEmotions com o auxílio da biblioteca itranslate [3] que facilita a utilização da API (*Application Programming Interface*) do Google Translate[4].

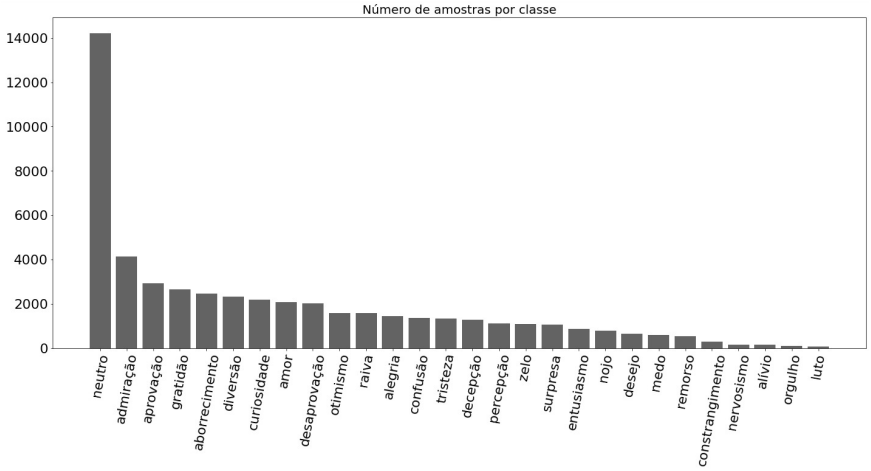


Figura 1. Distribuição das amostras por classe de emoção no *dataset* GoEmotions. Fonte: Própria

Sobre a distribuição das quantidades de exemplos no *dataset* GoEmotions (Figure 1), é possível notar um grande desbalanceamento. A classe mais frequente "neutro", tem aproximadamente 184 vezes mais amostras do que a classe "luto" que é a menos frequente. Caso nada for feito para amenizar este problema, o modelo classificador desen-

[3]https://github.com/ffreemt/google-itranslate

[4]https://translate.google.com.br/

volverá um viés, que resultará em uma habilidade preditiva ruim nas classes com poucos exemplos [Zheng and Jin 2020]. Para contornar esse problema utilizamos um método de balanceamento que será detalhado a seguir.

3.1. Modelo

O modelo BERT (*Bidirectional Encoder Representations from Transformers*) [Devlin et al. 2018], mais especificamente a versão pré-treinada para o português BERTimbau [Souza et al. 2020], foi escolhido para esta tarefa, pois modelos baseados na arquitetura *Transformers* [Vaswani et al. 2017], como o BERT, têm apresentado desempenho estado da arte nas mais diversas tarefas de PLN [Gillioz et al. 2020].

Como o *dataset* GoEmotions é *multi-label*, ou seja, pode ter um ou mais *labels* para cada exemplo, codificamos todos os *labels* atribuídos a cada sentença em um vetor *one-hot* e transformamos a tarefa em um problema de classificação binária para cada classe do modelo. Foi adicionada uma camada linear sobre o *pooled output* do modelo com um tamanho de saída igual a 28, equivalente ao número de classes do *dataset*. Utilizamos *Sigmoid* como função de ativação e *Binary Cross Entropy* como função de *loss*.

Utilizamos o método *Class Balanced Loss* (**CB**) para contornar o problema de desbalanceamento do *dataset* [Cui et al. 2019] . Neste método são calculados pesos para a função de *loss* ($\mathcal{L}$) utilizada no modelo, com base no número efetivo de amostras para cada classe. Um hiperparâmetro $\beta = 0.999$, também é utilizado para o cálculo destes pesos. Uma vez que, boa parte dos experimentos de [Cui et al. 2019] obteve um bom desempenho com este valor. A nova função que computa o *loss* do modelo é obtida por meio da seguinte equação:

$$\mathbf{CB}(p, y) = \frac{1 - \beta}{1 - \beta^{n_y}} \mathcal{L}(p, y), \tag{1}$$

onde p são as probabilidades fornecidas pelo modelo e n_y é o número de amostras presentes na classe y.

3.2. Fine-tuning

Fine-tuning é uma técnica de *transfer learning* na qual partindo de um modelo pré-treinado em uma tarefa de amplo domínio, os parâmetros desse modelo são ajustados para uma tarefa específica. Neste trabalho realizaremos o *fine-tuning* do BERTimbau-base e BERTimbau-large para a tarefa de CE no nível de sentenças. Os hiperparâmetros utilizados no *fine-tuning* do modelo estão listados na Tabela 1. Por limitação do hardware disponível, foi feito o *fine-tuning* do modelo BERTimbau-base em GPU e BERTimbau-large em CPU.

4. Análise dos Resultados

Para avaliar o desempenho do modelo foram utilizadas as métricas Precisão (P), Sensibilidade (S) e Medida-F (F1), mesmas métricas utilizadas pelos autores do *dataset* GoEmotions [Demszky et al. 2020], os quais utilizam em seus experimentos o modelo BERT-base para o inglês [Devlin et al. 2018]. Selecionamos apenas as predições do modelo com um *score* igual ou superior a 0,3. A Tabela 2 apresenta a comparação entre os resultados reportados por [Demszky et al. 2020] e pelo modelo BERTimbau-base e BERTimbau-large com o método de balanceamento, na tarefa de CE de 27 emoções no nível de sentença.

Hiperparâmetros	BERTimbau-base	BERTimbau-large
epochs	4	4
batch size	16	32
β (beta)	0.999	0.999
maximum sequence length	128	128
optimizer	*AdamW*	*AdamW*
learning rate	2e-5	2e-5
learning rate scheduler	*linear with warmup*	*linear with warmup*
warmup proportion	0.2	0.2

Tabela 1. Hiperparâmetros utilizados no *fine-tuning*.

Emoções	BERT-base EN [Devlin et al. 2018]			BERTimbau-base PT [Souza et al. 2020] Balanceado			BERTimbau-large PT [Souza et al. 2020] Balanceado		
	P	S	F1	P	S	F1	P	S	F1
admiração	0,53	0,83	0,65	0,58	0,75	0,66	0,60	0,75	0,67
diversão	0,70	0,94	0,80	0,76	0,89	0,82	0,75	0,91	0,82
raiva	0,36	0,66	0,47	0,37	0,46	0,41	0,40	0,49	0,44
aborrecimento	0,24	0,63	0,34	0,37	0,33	0,35	0,35	0,38	0,36
aprovação	0,26	0,57	0,36	0,40	0,40	0,40	0,42	0,40	0,41
zelo	0,30	0,56	0,39	0,34	0,44	0,39	0,34	0,47	0,39
confusão	0,24	0,76	0,37	0,33	0,56	0,41	0,35	0,58	0,44
curiosidade	0,40	0,84	0,54	0,44	0,77	0,56	0,45	0,80	0,57
desejo	0,43	0,59	0,49	0,53	0,55	0,54	0,60	0,58	0,59
decepção	0,19	0,52	0,28	0,28	0,20	0,23	0,28	0,25	0,26
desaprovação	0,29	0,61	0,39	0,37	0,43	0,40	0,39	0,45	0,42
nojo	0,34	0,66	0,45	0,46	0,42	0,44	0,46	0,47	0,46
constrangimento	0,39	0,49	0,43	0,39	0,38	0,38	0,39	0,35	0,37
entusiasmo	0,26	0,52	0,34	0,36	0,43	0,39	0,35	0,50	0,41
medo	0,46	0,85	0,60	0,54	0,73	0,62	0,57	0,77	0,65
gratidão	0,79	0,95	0,86	0,88	0,91	0,90	0,88	0,92	0,90
luto	0,00	0,00	**0,00**	0,13	0,67	**0,22**	0,17	0,50	0,25
alegria	0,39	0,73	0,51	0,51	0,57	0,54	0,51	0,60	0,55
amor	0,68	0,92	0,78	0,72	0,87	0,79	0,70	0,85	0,77
nervosismo	0,28	0,48	0,35	0,29	0,48	0,36	0,24	0,43	0,31
neutro	0,56	0,84	0,68	0,64	0,70	0,67	0,64	0,70	0,67
otimismo	0,41	0,69	0,51	0,52	0,56	0,54	0,53	0,53	0,53
orgulho	0,67	0,25	0,36	0,39	0,44	0,41	0,36	0,50	0,42
percepção	0,16	0,29	0,21	0,37	0,12	0,18	0,34	0,21	0,26
alívio	0,50	0,09	0,15	0,14	0,27	0,18	0,18	0,55	0,27
remorso	0,53	0,88	0,66	0,51	0,88	0,64	0,54	0,88	0,67
tristeza	0,38	0,71	0,49	0,47	0,54	0,50	0,48	0,57	0,52
surpresa	0,40	0,66	0,50	0,49	0,60	0,54	0,46	0,58	0,51
média macro	0,40	0,63	**0,46**	0,45	0,55	**0,48**	0,45	0,57	0,50

Tabela 2. Comparativo entre os modelos para inglês e português.

Podemos observar que ao utilizarmos a média macro da métrica Medida-F como parâmetro de avaliação do desempenho geral dos modelos analisados, o modelo para o português (BERTimbau-base), que tem a mesma quantidade de parâmetros do modelo utilizado para o inglês (BERT-base), apresentou desempenho superior. É possível atribuir isso ao método de balanceamento utilizado (*Class Balanced Loss*), já que em outros experimentos realizados não se utilizou o método de balanceamento e o desempenho das classes com a menor quantidade de exemplos foi consideravelmente prejudicado (Tabela 3). O maior ganho foi na classe "luto" que contém a menor quantidade de exemplos no *dataset*, passando de 0,00 para 0,22 na Medida-F. Um ponto importante é que o modelo obtém ótimos resultados nas classes "gratidão", "diversão" e "amor", acreditamos que isso ocorre devido a existência de palavras ou de expressões que contribuem fortemente para a identificação dessas emoções como "obrigado", "te amo/eu amo", ou gírias como "*lol, lmao, lmfao*" que estão presentes na maioria das sentenças relacionadas a essas emoções e dificilmente ocorrem em exemplos rotulados com outras emoções.

Outro ponto importante que deve ser levado em consideração é a qualidade de tradução do *dataset*. Por exemplo, a frase: "It's a better option because it's my life and none of your business? Lmfao, who are you", presente na base de treinamento, foi traduzida para: "É uma opção melhor porque é minha vida e nenhum da sua empresa? Lmfao, quem é você" , é possível perceber que na expressão idiomática "none of your business" foi realizada uma tradução literal dos seus termos, uma melhor tradução seria utilizar outra expressão idiomática com sentido equivalente para a língua portuguesa, como: "não é da sua conta". Esse tipo de problema leva a uma deterioração, ou completa perda, do sentido completo expresso pelas sentenças, podendo ser o suficiente para que a frase traduzida expresse uma emoção diferente, ou até mesmo, nenhuma emoção.

5. Conclusões

Por fim, podemos concluir que é necessária a criação de um *dataset* anotado, com uma boa variedade de emoções, no nível de sentença a ser utilizado na tarefa de CE em português, visto que, não encontramos esse tipo de recurso disponível na literatura.

Ainda, percebemos que a utilização do método de balanceamento e o modelo em português BERTimbau-base no *dataset* GoEmotions traduzido obteve melhores resultados se compararmos com o modelo em inglês no *dataset* GoEmotions original.

Como trabalho futuro pretendemos investigar o uso de diferentes ferramentas de tradução automática capazes de identificar e traduzir corretamente expressões idiomáticas, pois a qualidade da ferramenta de tradução utilizada deve impactar significativamente o desempenho dos modelos. Além disso, pretendemos construir um novo *dataset* anotado na mesma taxonomia de emoções empregada na construção do *dataset* GoEmotions, mas com sentenças provenientes de falantes da língua portuguesa. Pretendemos utilizar esses dados para verificar quanto desempenho é retido pelos modelos treinados com dados traduzidos, quando avaliados em sentenças originalmente em português.

Todos os códigos necessários para *download*, tradução do *dataset* e *fine-tuning* dos modelos utilizados estão disponíveis no GitHub e podem ser acessados através do link `https://github.com/Luzo0/GoEmotions_portuguese`.

Emoções	BERTimbau-base PT [Souza et al. 2020] Não Balanceado		
	P	S	F1
admiração	0,59	0,74	0,66
diversão	0,76	0,87	0,81
raiva	0,40	0,45	0,42
aborrecimento	0,36	0,32	0,34
aprovação	0,39	0,41	0,40
zelo	0,40	0,41	0,41
confusão	0,44	0,51	0,47
curiosidade	0,48	0,73	0,58
desejo	0,60	0,42	0,50
decepção	0,36	0,20	0,26
desaprovação	0,39	0,42	0,41
desgosto	0,52	0,41	0,46
constrangimento	0,00	0,00	**0,00**
entusiasmo	0,44	0,36	0,40
medo	0,55	0,76	0,64
gratidão	0,91	0,90	0,90
luto	0,00	0,00	**0,00**
alegria	0,52	0,56	0,54
amor	0,70	0,83	0,76
nervosismo	0,00	0,00	**0,00**
otimismo	0,57	0,57	0,57
orgulho	0,00	0,00	**0,00**
percepção	0,41	0,12	0,19
alívio	0,00	0,00	**0,00**
remorso	0,53	0,84	0,65
tristeza	0,44	0,56	0,49
surpresa	0,51	0,54	0,53
neutro	0,64	0,72	0,67
média macro	0,43	0,45	**0,43**

Tabela 3. Resultados sem o método de balanceamento.

Referências

Cowen, A., Sauter, D., Tracy, J. L., and Keltner, D. (2019a). Mapping the passions: Toward a high-dimensional taxonomy of emotional experience and expression. *Psychological Science in the Public Interest*, 20(1):69–90.

Cowen, A. S., Elfenbein, H. A., Laukka, P., and Keltner, D. (2019b). Mapping 24 emotions conveyed by brief human vocalization. *American Psychologist*, 74(6):698–712.

Cowen, A. S. and Keltner, D. (2017). Self-report captures 27 distinct categories of emotion bridged by continuous gradients. *National Academy of Sciences*, 114(38):7900–7909.

Cowen, A. S. and Keltner, D. (2020). What the face displays: Mapping 28 emotions conveyed by naturalistic expression. *American Psychologist*, 75(3):349.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277.

Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., and Ravi, S. (2020). Goemotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040—-4054. ACL.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosciatti, M. M., Ferreira, L., and Paraiso, E. (2013). Identificando emoçoes em textos em português do brasil usando máquina de vetores de suporte em soluçao multiclasse. *ENIAC-Encontro Nacional de Inteligência Artificial e Computacional. Fortaleza, Brasil*.

Duarte, L., Macedo, L., and Oliveira, H. G. (2019). Exploring emojis for emotion recognition in portuguese text. In *Proceedings of the EPIA Conference on Artificial Intelligence*, pages 719–730. Springer.

Ekman, P. (1992). An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Ekman, P. (2004). Emotions revealed. *Bmj*, 328(Suppl S5).

Gillioz, A., Casas, J., Mugellini, E., and Abou Khaled, O. (2020). Overview of the transformer-based models for nlp tasks. In *Proceedings of the 15th Conference on Computer Science and Information Systems*, pages 179–183. IEEE.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Pereira, D. A. (2021). A survey of sentiment analysis in the portuguese language. *Artificial Intelligence Review*, 54(2):1087–1115.

Plutchik, R. (1982). A psychoevolutionary theory of emotions. *Social Science Information*, 21(4-5):529–553.

Plutchik, R. (2003). *Emotions and life: Perspectives from psychology, biology, and evolution.* American Psychological Association.

Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: pretrained bert models for brazilian portuguese. In *Proceedings of the Brazilian Conference on Intelligent Systems*, pages 403–417. Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the Advances in neural information processing systems*, pages 5998–6008.

Wood, I. and Ruder, S. (2016). Emoji as emotion tags for tweets. In *Proceedings of the Emotion and Sentiment Analysis Workshop LREC2016, Portorož, Slovenia*, pages 76–79.

Zheng, W. and Jin, M. (2020). The effects of class imbalance and training data size on classifier learning: an empirical study. *SN Computer Science*, 1(2):1–13.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Sentiment Analysis in Portuguese Texts from Online Health Community Forums: Data, Model and Evaluation

Yohan Bonescki Gumiel[1,2], Isabela Lee[3], Tayane Arantes Soares[3], Thiago Castro Ferreira[1], Adriana Pagano[3]

[1]Departamento de Ciência da Computação, ICEX,
Universidade Federal de Minas Gerais
Belo Horizonte, MG – Brazil

[2]Pontifícia Universidade Católica do Paraná
Curitiba, PR – Brazil

[3]Faculdade de Letras,
Universidade Federal de Minas Gerais
Belo Horizonte, MG – Brazil

yohan.gumiel@pucpr.br

{isabelalee19,arantessoares}@gmail.com

{thiagocf05,apagano}@ufmg.br

Abstract. *This study introduces novel data and models for the task of Sentiment Analysis in Portuguese texts about Diabetes Mellitus. The corpus contains 1290 posts retrieved from online health community forums in Portuguese and annotated by two annotators according to 3 sentiment categories (e.g. Positive, Neutral and Negative). Evaluation of traditional (Support Vector Machine, Decision Tree, Random Forest and Logistic Regression classifiers) and state-of-the-art (BERT-based models) machine learning classifiers for the task showed the advantage in performance of the latter models as expected. Data and models are available to the community upon request.*

Resumo. *Este estudo apresenta dados e modelos para a Análise de Sentimentos de textos em português sobre Diabetes Mellitus. O corpus é composto por 1290 posts, extraídos de forums online sobre tópicos de saúde e anotados por dois estudandes de acordo com 3 categorias (e.g. Positivo, Neutro e Negativo). A avaliação de classificadores de Aprendizagem de Máquina (classificadores Support Vector Machine, Decision Tree, Random Forest e Logistic Regression) tradicionais e estado-da-arte (modelos baseados em BERT) mostrou a vantagem em performance do segundo tipo como esperado. Os dados e modelos estão disponíveis para a comunidade por meio de solicitação.*

1. Introduction

Social media provide a high volume of opinionated data from several sources, such as social networks, forums, and user reviews [Liu 2012]. Among Natural Language Processing (NLP) research concerns, little attention was paid to sentiment analysis until the

early 2000s [Yue et al. 2019]. The growth of social media concurs with advances in sentiment analysis research, this topic having become central in social media-related research nowadays [Liu 2012, Yadav and Vishwakarma 2020]. Sentiment analysis benefits from opinionated data to analyze the sentiments (attitudes, feelings, and opinions) towards entities [Yadav and Vishwakarma 2020]. Among a myriad of social media texts there are forums dedicated to particular topics regarding healthcare, which have become increasingly popular in recent years and which are worth exploring as sources of insight for strategies to reduce inequalities and promote well-being, two relevant goals in the UN 2030 Agenda for Sustainable Development.

People may not have ready access to primary care centers for preventive exams or they may find health professionals are not always available to provide adequate and detailed answers to all patients' questions or provide guidance to the patient and their families about all essential aspects of patient care during the consultation time [Gabarron et al. 2018]. Thus, patients often use social media to ask health-related questions and share experiences with other users with the same condition. Our study focuses on one such use of social media, namely online open-access communities. In particular, it draws on posts extracted from online health forums on diabetes, an increasingly prevalent severe and long-term condition affecting people worldwide and rapidly growing in Brazil. According to the most recent survey, dated from 2019, it is estimated that almost half a billion people (9.3% of the adults between 20-79 years) were living with diabetes in 2019 [Saeedi et al. 2019]. Most alarmingly, one every two (50.1%) persons living with diabetes is not aware or has not been diagnosed as having diabetes. Hence, studies that aim to explore the language used in social media by people who seek advice on diagnosed or suspected diabetes are expected to contribute to devising healthcare tools for better support and prevention. Our study seeks to develop a framework to automatically detect social media user sentiment modelled upon data from diabetes online health forum posts. This framework is scheduled to be integrated with other applications, such as chatbots, to control the bot dialog flow according to the user's sentiment.

Prior studies analyzed online health communities' posts, tweets, and news-related comments for the diabetes context. [Gabarron et al. 2019, Salas-Zárate et al. 2017, Cignarelli et al. 2020] analyzed diabetes-related tweets, whereas [Liu et al. 2020, Lu et al. 2017] drew on online health communities' posts. Further, the study by [Liu et al. 2018] examined opinion news about diabetes and their respective comments. None of these studies, however, has addressed social media posts in online communities in Brazil. There is a lack of social media datasets and studies for sentiment analysis for the Portuguese language considering the healthcare domain. Hence, we purported to fill this gap by annotating a sentiment analysis dataset and evaluating the sentiment extraction results using traditional and state-of-the-art machine learning classifiers.

Related Work The study by [Liu et al. 2018] noticed that positive emotions were expressed more frequently than negative sentiments when analyzing the sentiment over an online health community about diabetes. However, the posts collected for our corpus had a different characteristic, being generally more negative or neutral.

	Sentiment	Post
#1	Positive	Pessoal, obrigado para quem orou e para quem fez oração. Meu médico me disse que não sabe como, mas eu fiz meus exames e está tudo ótimo. Graças a Deus. Tirou até a insulina e vou tomar apenas um Glifage. Obrigada, Deus. EMOJIS *Folks, thank you to those of you who prayed for me. My doctor told me he's got no explanation for this, I got my test results and everything is fine. Thank God. He even took insulin out of my prescription and I'm only taking a Glifage pill. Thank you, Lord. EMOJIS*
#2	Neutral	O que vocês acham dos derivados do leite? Por exemplo: queijo, manteiga, iogurte. Faz bem ou mal para a saúde do diabético? *What do you think about dairy products? For example: cheese, butter, yoghurt. Are they good or bad for diabetic people's health?*
#3	Negative	Estou com minha unha do dedão esbranquiçada e dolorida. Está muito fraca e quebradiça. Alguém sabe de algo que posso usar? *My toenail is whitish and sore. It is very weak and brittle. Can anyone suggest anything for me to try?*
#4	Neutral	Podemos comer amendoim japonês? *Can we eat Japanese-style peanuts?*
#5	Neutral	Posso fazer o suco de chuchu, sem limão? *Can I prepare chayote juice with no lemon?*
#6	Negative	Sensação doce na boca, mesmo sem comer doces. Por que será? *Sweet aftertaste in my mouth, even when I haven't had any sweets. How come?*

Table 1. Examples of posts with annotated sentiments and free-translation into English

2. Corpus

Material We collected a total of 1290 texts from open Social Media forums on diabetes management and self-care in Brazil, encompassing communities with over 80 thousand Portuguese speaking users.

Annotators Annotation of the texts was performed by two undergraduate students in Language and Arts.

Anonymization To ensure users' privacy and anonymity, before the annotation starts, the selected texts were de-identified by first removing some emojis, fixing orthographic mistakes and paraphrasing non-fluent syntactic structures. Then any identifier, such as name, phone or address, was removed from the questions. Moreover, *quasi-identifiers*, such as age and relative mentions, were modified. Users' age were modified by randomly choosing a number in the interval of $[age - 5; age + 5]$, whereas mentions to relatives were randomly changed by the reference to a relative with similar age, such as *parent* $\leftrightarrow$ *uncle, parent in law*; *sibling* $\leftrightarrow$ *cousin, partner*; *son* $\leftrightarrow$ *nephew*.

Annotation The texts were annotated with sentiment polarity according to three categories: Positive, Negative and Neutral. The annotations were conducted with Label Studio, an open-source web annotation tool [Tkachenko et al. 2021].

The annotators were requested to consider the overall sentiment polarity in each post bearing in mind semantic and pragmatic features of the texts. Exclamation marks were considered intensification of emotion, leading to texts that contained those punctuation marks being annotated as positive or negative. As the domain of the texts pertains to a chronic health condition, negative polarity was expected to be dominant. In view of this, a more fine-grained definition of polarity categories was needed so that features indicating positive and neutral polarity could be detected in the analysis. Hence the 3 polarity categories were defined as:

- *Positive*: statements by users in the first person with references to their experience regarding personal achievement and counseling others towards adoption of healthy habits and promotion of well-being
- Negative: statements by users in the first person with references to their experience regarding symptoms and difficulties in managing diabetes
- *Neutral*: objective statements, with no reference to personal experience, pertaining to seeking general information, frequently on food and medication

By "references to experience", we mean instances in which the user states his/her personal opinion and explicitly describes his/her feelings in terms of physical and mental state.

Examples of the three sentiment categories are illustrated in examples #1, #2, and #3 in Table 1. Some posts demanded closer inspection to annotate their sentiment. For instance, texts with 1st person singular or plural pronominal forms but not construing any personal meaning were classified as Neutral. Examples #4 and #5 in Table 1 illustrate such case: the posts are written by users seeking information on food. Both use a 1st

Sentiment	Annotations	Avg. Characters	Avg. Tokens
Positive	44	156.32	27.57
Neutral	575	71.3	12.35
Negative	671	146.27	25.54
All	1290	113.19	19.73

Table 2. Corpus Statistics

Model	Precision	Recall	F1-Score
Decision Tree	63.92 (±0.01)	65.16 (±0.01)	64.43 (±0.01)
Random Forest	74.67 (±0.00)	76.43 (±0.00)	75.01 (±0.00)
Logistic Regression	72.95 (±0.00)	75.58 (±0.00)	74.24 (±0.00)
SVM	70.79 (±0.00)	73.26 (±0.00)	71.95 (±0.00)
mBERT	77.62 (±0.02)	77.52 (±0.02)	76.79 (±0.02)
BioBERTpt	78.56 (±0.02)	77.29 (±0.02)	76.23 (±0.02)
BERTimbau	**83.12** (±0.01)	**82.67** (±0.01)	**82.53** (±0.01)

Table 3. Classifiers performance considering weighted precision, recall and F1-score, averaged after 10 runs.

person pronoun, but they do not refer to the users' personal experiences. Thus, they were labelled Neutral. In contrast, example #6 in Table 1 illustrates a post which was annotated as Negative, even though no first person pronoun is used in Portuguese. The post construes an individual experience linked to a common diabetes symptom, namely a sweet aftertaste, a likely warning sign that the person may have diabetes.

Inter-annotator Agreement Using Cohen-kappa [Cohen 1960], we computed an inter-annotator agreement value of 0.64, which can be considered as a substantial agreement by the scale provided by [Landis and Koch 1977]. The annotators took on average 47.71 seconds to classify the sentiment in each text.

Corpus Our corpus statistics are shown in Table 2. From our 1290 annotated texts, most of them had Neutral and Negative annotations, with a total of 575 (44.57%) Neutral sentiment annotations and 671 (52.02%) Negative sentiment annotations. There were few Positive sentiment annotations in the corpus, totalling 44 (3.41%) annotations. With regard to average number of characters and tokens, Positive annotations and Negative ones had similar numbers, whereas Neutral annotations had a smaller average number of characters and tokens. This may suggest that Neutral posts tend to be shorter than Positive and Negative ones.

3. Models

We evaluated several traditional and state-of-the-art classifiers for sentiment analysis in the domain of Diabetes. As traditional ones, we performed experiments with Support Vector Machines (SVM), Random Forest (RF), Decision Tree (DT) and Logistic Regression (LR) classifiers. As input features to the classic classifiers, we used the TF-IDF vector representations of the texts to be classified.

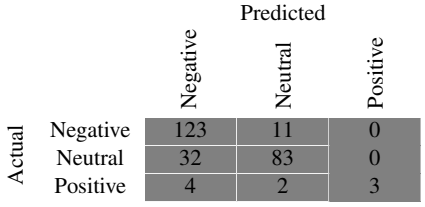


Figure 1. Confusion matrix for the BERTimbau model.

As state-of-the-art approaches, we fine-tuned BERT-based models such as its multilingual version [Devlin et al. 2019]; BERTimbau [Souza et al. 2020], its Brazilian Portuguese version; and BioBERTpt [Schneider et al. 2020], a Brazilian Portuguese version of BERT focused on the clinical domain.

4. Experiment

Data To evaluate the approaches, our collected corpus was divided into training set, development set and test set. We selected 774 texts (60%) for the training set, 258 (20%) for development set and 258 (20%) for testing set, maintaining the class stratification among the splits.

Method The sentiment polarity extraction was treated as a multi-class classification task, where each model was trained to predict one out of the three sentiment categories (e.g., Positive, Neutral or Negative) given the input text. Models were evaluated by computing the weighted Precision, Recall and F-Score in the test set, averaged after 10 runs.

5. Results

Results of the model of sentiment analysis in the medical domain are summarized in Table 3. Traditional machine learning classifiers did not perform as well as the contextual models, as expected. From the contextual models, the fine-tuning of BERTimbau achieved the best weighted F1-score, being able to infer correct predictions for both minority and majority classes. BERTimbau was followed by mBERT and BioBERTpt, with both having similar performance, with sightly superior results for mBERT.

Aiming to verify the label distribution over the predictions for our best model (BERTimbau), we provide a confusion matrix in Figure 1. The BERTimbau model was not that effective in predicting Neutral and Positive classes. Examples that had a Neutral annotation were miss-classified as Negative several times. Further, due to the lack of Positive examples, the model incorrectly predicted examples with Positive annotations as Negative and Neutral annotations.

6. Discussion

This study introduces novel data and models for the task of Sentiment Analysis in Portuguese texts about Diabetes Mellitus.

Corpus We had our texts annotated according to three sentiment categories: Positive, Neutral and Negative. There was a smaller amount of Positive annotations in comparison to Negative and Neutral annotations. In general, users tended to have a negative or neutral sentiment when asking questions or talking about their problems, primarily because of their anxiety or uncertainty about the related topic. According to the study by [Gabarron et al. 2019], which analyzed sentiments over diabetes-related tweets, they noticed that tweets about Type 2 Diabetes were more negative, particularly when there were no emojis. For privacy reasons, many of the emojis in the texts of our corpus were removed and perhaps this may be a bottleneck from our corpus. On the other hand, we judged the anonymization process as necessary in order to respect privacy of the users and in agreement with ethical concerns.

In addition, [Gabarron et al. 2019] noticed that tweets about Type 2 Diabetes were more negative than tweets about Type 1 Diabetes. In our corpus we made no distinction about Diabetes type; however, diabetes type could be an influencing factor over post negativity prevalence.

Classifiers We trained both traditional and state-of-the art machine learning classifiers for the target task. As expected, the current state-of-the-art approaches based on contextual models trained over large corpora achieved overall superior results. Further, the pre-trained models specific to Portuguese (BioBERTpt and BERTimbau) showed improved performance over mBERT, which is pre-trained over several languages. Additionally, the BERTimbau, which is pre-trained over a large and diverse corpus of internet pages, had superior performance compared to BioBERTpt, which is initialized with mBERT and trained on clinical notes and biomedical literature. The fine-tuning of BERTimbau had the f1-scores on micro, macro, and weighted evaluation, showing its ability to predict both majority and minority classes.

7. Conclusion

We found diabetes posts in social media a sensitive context, with few positive posts. Hence, predicting positive sentiments became a challenge to the classifiers. However, the state-of-the-art models, especially the fine-tuning of BERTimbau, were able to overcome that challenge.

Contextual models that were specific to the Portuguese language achieved overall superior performance, showing the benefit of developing models that are customized to a particular language.

As future work, we plan to expand our corpus, especially the number of positive annotations. Further, we plan to expand this study by including posts on other chronic conditions or diseases. Additionally, we plan to provide further experiments with traditional machine learning classifiers with ablation tests to verify the effect of other features besides representations from TF-IDF.

References

Cignarelli, A., Sansone, A., Caruso, I., Perrini, S., Natalicchio, A., Laviola, L., Jannini, E. A., and Giorgino, F. (2020). Diabetes in the time of covid-19: A twitter-based sentiment analysis. *Journal of Diabetes Science and Technology*, 14(6):1131–1132.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Gabarron, E., Bradway, M., Fernandez-Luque, L., Chomutare, T., Hansen, A. H., Wynn, R., and Årsand, E. (2018). Social media for health promotion in diabetes: study protocol for a participatory public health intervention design. *BMC Health Services Research*, 18(1):414.

Gabarron, E., Dorronzoro, E., Rivera-Romero, O., and Wynn, R. (2019). Diabetes on twitter: A sentiment analysis. *Journal of Diabetes Science and Technology*, 13(3):439–444.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan Claypool Publishers.

Liu, X., Sun, M., and Li, J. (2018). Research on gender differences in online health communities. *International Journal of Medical Informatics*, 111:172–181.

Liu, Y., Stouffs, R., and Theng, Y. L. (2020). Sentiment analysis on social media for identifying public awareness of type 2 diabetes. In *The 54th International Conference of the Architectural Science Association (ANZAScA)*.

Lu, Y., Wu, Y., Liu, J., Li, J., and Zhang, P. (2017). Understanding health care social media use from different stakeholder perspectives: A content analysis of an online health community. *J Med Internet Res*, 19(4):e109.

Saeedi, P., Petersohn, I., Salpea, P., Malanda, B., Karuranga, S., Unwin, N., Colagiuri, S., Guariguata, L., Motala, A. A., Ogurtsova, K., Shaw, J. E., Bright, D., and Williams, R. (2019). Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the international diabetes federation diabetes atlas, 9th edition. *Diabetes Research and Clinical Practice*, 157:107843.

Salas-Zárate, M. d. P., Medina-Moreira, J., Lagos-Ortiz, K., Luna-Aveiga, H., Rodríguez-García, M. Á., and Valencia-García, R. (2017). Sentiment analysis on tweets about diabetes: An aspect-level approach. *Computational and Mathematical Methods in Medicine*, 2017:5140631.

Schneider, E. T. R., de Souza, J. V. A., Knafou, J., Oliveira, L. E. S. e., Copara, J., Gumiel, Y. B., Oliveira, L. F. A. d., Paraiso, E. C., Teodoro, D., and Barra, C. M. C. M. (2020). BioBERTpt - a Portuguese neural language model for clinical named entity recognition. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 65–72. Association for Computational Linguistics.

Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: Pretrained bert models for brazilian portuguese. In Cerri, R. and Prati, R. C., editors, *Intelligent Systems*, pages 403–417, Cham. Springer International Publishing.

Tkachenko, M., Malyuk, M., Shevchenko, N., Holmanyuk, A., and Liubimov, N. (2020-2021). Label Studio: Data labeling software. Open source software available from https://github.com/heartexlabs/label-studio.

Yadav, A. and Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385.

Yue, L., Chen, W., Li, X., Zuo, W., and Yin, M. (2019). A survey of sentiment analysis in social media. *Knowledge and Information Systems*, 60(2):617–663.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

A Weakly Supervised Dataset of Fine-Grained Emotions in Portuguese

Diogo Cortiz[1,2], Jefferson O. Silva[2,4], Newton Calegari[2], Ana Luísa Freitas[3], Ana Angélica Soares[3], Carolina Botelho[3], Gabriel Gaudencio Rêgo[3], Waldir Sampaio[3], Paulo Sergio Boggio[3]

[1]Brazilian Network Information Center (NIC.br)
São Paulo, SP – Brazil

diogo@nic.br

[2]Pontifical Catholic University of São Paulo (PUC-SP)
São Paulo, SP – Brazil

{dcortiz, silvajo, njcalegari}@pucsp.br

[3]Mackenzie Presbyterian University
São Paulo, SP – Brazil

{paulo.boggio}@mackenzie.br

[4]Jusbrasil
São Paulo, SP – Brazil

Abstract. *Affective Computing is the study of how computers can recognize, interpret and simulate human affects. Sentiment Analysis is a common task in NLP related to this topic, but it focuses only on emotion valence (positive, negative, neutral). An emerging approach in NLP is Emotion Recognition, which relies on fined-grained classification. This research describes an approach to create a lexical-based weakly supervised corpus for fine-grained emotion in Portuguese. We evaluate our dataset by fine-tuning a transformer-based language model (BERT) and validating it on a Gold Standard annotated validation set. Our results (F1-score= .64) suggest lexical-based weak supervision as an appropriate strategy for initial work in low resourced environment.*

Resumo. *A Computação Afetiva é o estudo de como os computadores podem reconhecer, interpretar e simular os afetos humanos. A Análise de Sentimento é uma tarefa comum em PLN, mas se concentra apenas na valência da emoção (positiva, negativa, neutra). Uma abordagem emergente é o Reconhecimento de Emoção, que depende de uma classificação refinada. Nesta pesquisa, descrevemos uma abordagem de supervisão fraca baseada em Itens Lexicais para criar um corpus de emoções refinadas em português. Avaliamos nosso corpus fazendo o ajuste fino de um modelo de linguagem baseado em Transformer (BERT) e avaliando-o em um conjunto de validação anotado. Nossos resultados (F1-score= .64) sugerem que a supervisão fraca baseada em Itens Lexicais pode ser uma estratégia apropriada para o trabalho inicial em ambiente de poucos recursos.*

1. Introduction

Affective Computing comprises the study of how computers can recognize, interpret and simulate human affects. According to Picard [Rosalind 2000], a field pioneer, it is imperative to develop ways for computers to be able to recognize, understand and express emotions for intelligent and natural interaction between humans and machines. Although Affective Computing may employ several input types such as facial expression images, voice, or physiological data, our research focuses on the written language. Thus, our scope lies in the area of Natural Language Processing (NLP).

A common task in NLP is Sentiment Analysis which classifies a text into three different categories: positive, negative, and neutral [Drus and Khalid 2019]. The Emotion Recognition task, however, is a more detailed NLP task. Rather than classifying text only into valence categories (positive, negative, or neutral), it classifies text into more detailed emotional categories.

The delimitation of this research is concentrated in the area of fine-grained Emotion Recognition. We studied an approach to create a corpus in Portuguese for this task using a weak supervision approach.

2. Related Work

Several works in NLP are based on the theory of basic emotions [Ekman 1992] to classify texts into defined categories of emotions, the number of categories ranging from 4 (four) to 8 (eight). One of the studies adopted the 6 (six) basic emotions proposed by Ekman (joy, fear, anger, sadness, surprise, and disgust) to train an Emotion Recognition model [Batbaatar et al. 2019]. Another research added trust and anticipation to the basic emotions, working with 8 (eight) basic emotion categories [Sosea and Caragea 2020].

In the realm of emotion study, other relevant theory is the Theory of Constructed Emotion [Barrett 2016], which assumes that emotions are not universal, but idiosyncratic. This theoretical debate imposes methodological limitations that a different computational approach may help to solve. The Semantic Space Theory [Cowen and Keltner 2021] let us recognize and analyze emotional content of naturalistic stimuli, using open-ended statistical techniques to capture emotional variations in behavior. The results suggest that more than 25 emotional classes have distinct profiles of previous expressions and events. The authors argue that these emotions are high-dimensional, categorical, and often blended.

GoEmotions is a dataset with more than 58,000 English Reddit comments for training NLP models in the Emotion Recognition task. It was annotated for 27 categories of emotions and Neutral based on the Semantic Space Theory. The authors fine-tuned a BERT language model and achieved an average F_1-score of $.46$ [Demszky et al. 2020].

Despite the average F_1-score below $.50$, some classes scored above $.70$. It is a complex dataset with many categories that often have fuzzy boundaries between them. It is essential to discuss the importance of creating a fine-grained dataset with more emotional categories. Research in the Affective Computing area is not limited to Sentiment Analysis or categories proposed by the basic theory.

Although the GoEmotion dataset was released according to open data standards [Demszky et al. 2020], the scope of the corpus is limited to English, which makes it difficult to use in applications in other languages. One of the challenges that Machine Learn-

ing faces is dealing with a low-resourced environment (when the data available is not enough to train the models). This phenomenon can happen in specific domains of applications but also in specific geographic regions.

In the area of NLP, there is a lack of datasets and corpus available in many languages. It is the case of Portuguese, which has a small amount of Sentiment Analysis datasets when compared to English [Pereira 2021]. It is worth noting that we searched for a dataset of fine-grained emotion, but we did not find any in Portuguese.

3. Objectives, Research Questions and Hypothesis

This research aims to study the creation of a corpus of fine-grained emotions for low resourced languages, specifically Portuguese. Due to limited financial resources, a specific objective of this work is to study the use of the weak supervision strategy to construct our corpus. Weak supervision is a strategy when there is no human annotation of each data point, but the labels are attributed using noisy and limited sources or specific rules. We proposed the following research questions (RQ) to guide our work:

RQ1: Is the weak supervision strategy suitable for building an NLP corpus for the fine-grained Emotion Recognition task in a low resourced environment?

RQ2: What is a proper weak supervision approach to construct a corpus for fine-grained Emotion Recognition tasks in NLP?

Our first hypotheses (H1) is that weak supervision could be a suitable strategy to build NLP corpus for emotion recognition. Our second hypotheses (H2) is that lexical-based approach can be an adequate strategy to collect samples for each of the categories of our dataset, using the Lexical Items (LI) as a criterion for defining the label in an adequate way for Portuguese. A third hypothesis (H3) is that using SOTA Machine Learning techniques (specifically Transformers-based language models), combined with masking techniques in the LI presented in the weakly supervised corpus, can avoid the model overfit the learning phase.

To answer RQ1 and RQ2 and validate our hypotheses, we prepared an experiment to create a weakly supervised corpus in Portuguese and measure its performance by training a classification model. The following sections will describe our experimental protocol, including how we collected and weakly annotated the data, our model architecture, metrics, and results.

4. Experimental Protocol

Our experiment is composed of the following pipeline: defining emotion categories based on semantic space theory for Portuguese; selecting the lexical items related to each emotion category based on its definition; collecting the data; manually annotating a test dataset to create a gold standard; defining the model architecture; training the model and evaluating it on the gold standard. Each of them is described in detail in this paper.

4.1. Defining Emotion Categories

The emotion categories for this research were defined from a review of the GoEmotion work [Demszky et al. 2020]. The review process had two stages and the participation of a

group of 7 (seven) researchers with different backgrounds (psychology, neuroscience, sociology, communications, cognitive science, and computer science). In the first stage, the researchers discussed and reviewed each emotion in English during a working meeting. They proposed a translation into Portuguese based on the definitions of each emotion. The result of this first stage was a translated list of terms with consensus among the reviewers.

The second stage was reviewing the categories' definitions in Portuguese to check if they were consistent with the language. The reviewers suggested changing the emotional category *cuidado*, translated from *caring* to *compaixão*, as it is a more broad and blended category in the Portuguese language. The second proposal was the removal of the emotion *realization*, in the sense of perceiving something, as it is not a much prevalent emotional category in the Portuguese language. Finally, there was a consensus among researchers to add the categories *saudade* and *inveja* to the list. We also removed neutral to focus on emotions. The final list consists of 28 emotional categories in total. All emotions and their definitions are presented in Table 1.

Table 1. Portuguese Emotion Categories

CATEGORY IN PORTUGUESE	CORRESPONDING IN GOEMOTIONS	DEFINITION IN PORTUGUESE
ADMIRAÇÃO	ADMIRATION	Achar algo impressionante ou digno de respeito.
DIVERSÃO	AMUSEMENT	Achar algo engraçado ou se divertir.
RAIVA	ANGER	Forte sentimento de desprazer ou antagonismo.
ABORRECIMENTO	ANNOYANCE	Raiva leve, irritação.
APROVAÇÃO	APPROVAL	Ter ou expressar uma opinião favorável.
CONFUSÃO	CONFUSION	Falta de compreensão, incerteza.
CURIOSIDADE	CURIOSITY	Forte desejo de saber ou aprender algo.
DESEJO	DESIRE	Forte sentimento de querer algo ou desejar que algo aconteça.
DECEPÇÃO	DISAPPOINTMENT	desprazer causado pelo não cumprimento de expectativas.
DESAPROVAÇÃO	DISAPPROVAL	Ter ou expressar opinião desfavorável.
NOJO	DISGUST	Repulsa despertada por algo desagradável ou ofensivo.
VERGONHA	EMBARRASSMENT	Vergonha ou constrangimento.
ENTUSIASMO	EXCITEMENT	Sensação de grande empolgação e ansiedade.
MEDO	FEAR	Estar com medo ou preocupado.
GRATIDÃO	GRATITUDE	Sentimento de gratidão e apreciação.
LUTO	GRIEF	Tristeza intensa, especialmente causada pela morte de alguém.
ALEGRIA	JOY	Sensação de prazer e felicidade.
AMOR	LOVE	Forte emoção positiva de consideração e carinho.
NERVOSISMO	NERVOUSNESS	Apreensão, preocupação, ansiedade.
OTIMISMO	OPTIMISM	Esperança sobre o futuro ou sobre o sucesso de algo.
ORGULHO	PRIDE	Prazer devido às próprias conquistas ou de alguém
ALÍVIO	RELIEF	Tranquilidade e relaxamento após ansiedade ou angústia.
REMORSO	REMORSE	Arrependimento ou sentimento de culpa.
TRISTEZA	SADNESS	Dor emocional, tristeza.
SURPRESA	SURPRISE	Sentir-se surpreso, assustado com algo inesperado.
INVEJA	-	Desgosto provocado pela felicidade ou prosperidade alheia
SAUDADE	-	Lembrança grata de pessoa ausente ou um momento passado.
COMPAIXÃO	-	Sentimento piedoso de simpatia e de ajuda

4.2. Selecting Lexical Items for weak supervision

After translating and defining emotions into Portuguese, the next step was to select the Lexical Items that would serve as a filter to search for examples and label assignment rules

(weak supervision). For each of the emotions on the list, we initially look for related Lexical Items by synonyms. For this, we use the database available at www.sinonimos.com.br, which has more than 30 thousand synonyms of words and expressions for Portuguese.

Because some words of emotion presented polysemic behavior, we opted for human curation to select the proper Lexical Items. Only synonyms with a semantic relationship with the definition of emotion were considered. For each Verbal Lexical Item, we collect the different conjugations in the repository www.conjugacao.com.br to cover all tenses and moods in Portuguese. To avoid the negation effect, we manipulated the data as follows: we searched for the combination of the word "não" (no/not in Portuguese) or "nem" (neither in Portuguese) followed by a Lexical Item in our list. If an example was found, we removed it from our dataset. We also added slang and terms related to emotions that were known to the authors. The result of this step was a list in which each emotion was associated with a set of lexical items, which were later used as a data collection filter and label assignment rule.

4.3. Data

We use Twitter as a data source. The collection was made between the 23rd and 24th of June (2021) using the platform's official API. The filters used were the list of terms associated with each emotion. Retweets and replies were not considered, keeping only original tweets. Hashtags were removed, but emojis were kept.

In total, 49179 tweets were collected using a weak supervision approach. Each example received the category label according to the Lexical Item used in the collection. For example, if a tweet was collected because it was filtered by a term associated with the emotion *amor*, it would be labeled to the *amor* category.

We tried to maintain a balanced distribution of examples among the classes, but the results of our collection process suggest that some emotional categories are more prevalent than others. We intend to focus on additional data collection for the categories with the smallest number of examples to achieve a better balance distribution in future work. For the training set, we had a total of 47405 examples. We present in Figure **??** the total number of examples by category and the descriptive statistics of our dataset [1].

4.3.1. Masking Lexical Itens

A hypothesis that appeared during the execution of this research was that the models could memorize the Lexical Items (LI) associated with each emotion, reducing generalization properties and causing the model to overfit. We chose to apply a masking technique to the Lexical Items used for collection and label assignment to investigate this phenomenon. The masking technique consisted of replacing an LI by [MASK], as can be seen in the examples in Table 2.

We ended up with three datasets for training three different models. The first is the original dataset that we created using the weakly supervised approach without any masking technique. We identified this dataset as NoMask. The second dataset is the result of applying the masking technique to 30% of examples for each category. We identified

[1]Data available at: https://github.com/diogocortiz/PortugueseEmotionRecognitionWeakSupervision

Figure 1. Examples per categories.

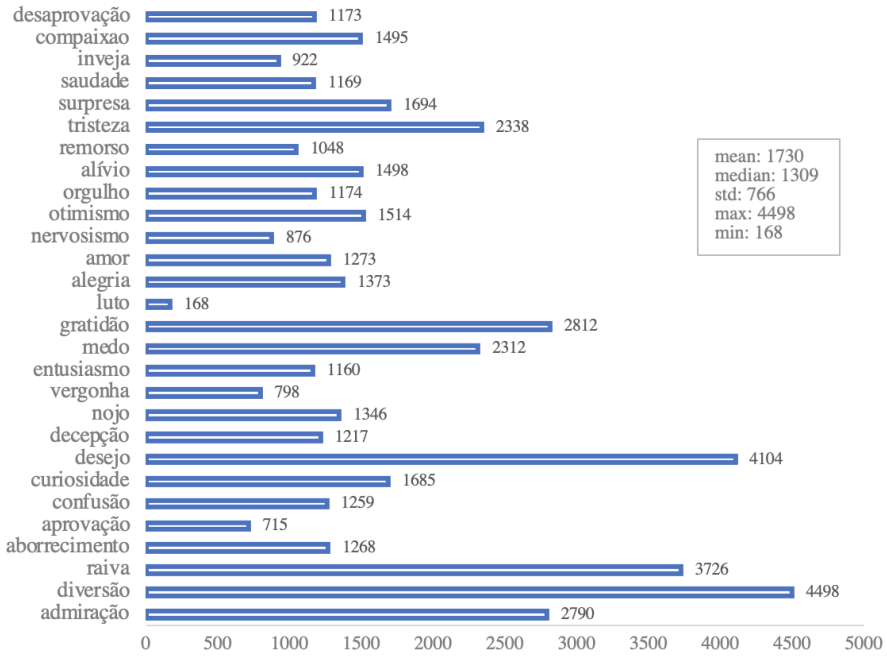


Table 2. Masking technique

Original	tô indignada e não é pouco!
Masked	tô [MASK] e não é pouco!

this dataset as 30Mask. The third dataset is the result of masking all Lexical Items. We identified this dataset as FullMasked.

4.3.2. Gold standard for validation

Despite this research studying the feasibility of the weak supervision approach for the Emotion Recognition task in NLP, it is worth noting the importance of building a dataset with human curation to evaluate the performance of a trained model with the created dataset.

To meet this requirement, we separated a set composed of 1773 examples from the dataset created earlier; we removed the labels assigned by the weak supervision approach so that a human could manually annotate them. We did not apply any Masking technique to this set. Due to limited resources, it was not possible to cross-annotate the validation dataset. Only one annotator annotated each example. For this reason, it is not possible to present any measure of agreement between the annotators. We recognize the limitations of this procedure, which can reduce the quality of supervision and introduce bias.

4.4. Models

To study the performance of our dataset, we needed to fine-tune the BERT language model to the Emotion Recognition task using our weakly supervised dataset. The Bidirectional Encoder Representations from Transformers (BERT) pre-trained language model [Devlin et al. 2018] was released by Google in 2018. Since then, the use of this architecture has improved the performance in different natural language tasks. In our research, we used BERTimbau [Souza et al. 2020], a pre-trained BERT model for Brazilian Portuguese. We fine-tuned three different models using our three different datasets (NoMask, 30Mask, and FullMask).

4.4.1. Parameters Settings

When finetuning the BERT language model, we keep most of the hyperparameters set in the original paper [Devlin et al. 2018]. We changed only batch size and learning rate as proposed by [Demszky et al. 2020]. We trained each model for 4 (four) epochs. The threshold to set a classification as positive was .30 (the same used by [Demszky et al. 2020]). All models were implemented using the huggingface library. The training process used the same computing environment (Quadro RTX 6000).

4.5. Results

The results in Table 3 show the performance of our three models. As we can observe, the model trained with 30% of LI masked (30Mask) had a similar performance to the model trained with the original dataset with no intervention (NoMask).

Those results suggest that masking a certain amount of Lexical items used as a label rule (weak supervision) could be an appropriate strategy to stimulate the model to learn by context and not only by memorizing LI. However, masking all LI introduces much noise to the training dataset, significantly impacting the model performance.

5. Conclusions

According to the results presented, we argue that the adoption of Weak Supervision may be an appropriate strategy for some NLP activities in low-resource scenarios. The creation of datasets is costly and often prohibitive for some economies, making weak supervision an initial alternative for projects when there are insufficient resources to adopt a human supervision methodology. Our RQ1 inquired whether weak supervision is a proper approach to construct a corpus for fined-grained Emotion Recognition in low resourced environment. We found consistent results when evaluating our models, suggesting that weak supervision is an appropriate approach for initial work in the Emotion Recognition NLP task in Portuguese. The results supports our first hypotheses (H1).

This research used a Lexical-based approach to collect, and weak supervise the dataset. According to the results achieved and based on our empirical experience during its execution, we argue that this approach can be appropriate for collecting and annotating data in tasks involving narrow scenarios and well-defined problems. The results help us to answer our RQ2 and validate the H2. However, our experiment has some limitations, such as the validation dataset created from the initial collection of Lexical Items, which

Table 3. Results based on weak supervision

	NoMask			30Mask			FullMask		
Emotion	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
admiração	.67	.44	.53	.71	.44	.54	.38	.39	.39
diversão	.49	.50	.49	.54	.50	.52	.23	.33	.27
raiva	.84	.50	.63	.80	.50	.62	.45	.15	.22
aborrecimento	.82	.74	.78	.81	.75	.78	.47	.26	.34
aprovação	.58	.38	.46	.60	.36	.45	.26	.11	.16
confusão	.68	.66	.67	.66	.63	.65	.42	.22	.29
curiosidade	.71	.61	.66	.71	.61	.66	.37	.15	.21
desejo	.52	.54	.53	.49	.51	.50	.15	.12	.13
decepção	.69	.40	.51	.71	.40	.51	.43	.05	.10
nojo	.81	.95	.87	.83	.97	.89	.53	.15	.24
vergonha	.88	.89	.88	.87	.86	.87	.29	.06	.10
entusiasmo	.74	.91	.81	.72	.89	.80	.30	.16	.21
medo	.75	.93	.83	.76	.87	.81	.43	.35	.38
gratidão	.31	.57	.40	.32	.57	.41	.09	.32	.14
luto	.91	.56	.69	.83	.56	.67	.00	.00	.00
alegria	.68	.66	.67	.69	.65	.67	.27	.24	.25
amor	.88	.50	.64	.80	.50	.61	.39	.41	.40
nervosismo	.94	.64	.76	.86	.66	.74	.50	.03	.06
otimismo	.49	.42	.45	.50	.45	.47	.20	.05	.09
orgulho	.70	.59	.64	.70	.59	.64	.10	.03	.04
alívio	.55	.89	.68	.54	.86	.67	.15	.22	.18
remorso	.64	.71	.67	.62	.71	.66	.40	.08	.13
tristeza	.71	.54	.62	.76	.48	.58	.47	.19	.27
surpresa	.61	.91	.73	.61	.85	.71	.48	.44	.46
saudade	.75	.68	.72	.75	.72	.74	.52	.32	.40
inveja	.93	.93	.93	.93	.93	.93	.88	.34	.49
compaixão	.58	.88	.70	.59	.88	.71	.25	.12	.16
desaprovação	.60	.02	.03	.50	.02	.03	.26	.03	.06
macro avg	**.70**	**.64**	**.64**	**.69**	**.63**	**.64**	**.35**	**.19**	**.22**

makes it difficult to assess the generalization performance of the models. In this sense, we can neither validate nor refute our third hypothesis (H3). We plan to build a new dataset with human supervision in future work without using the Lexical Items list in the filter during data collection. It will be possible to validate and compare the generalization performance of models using different datasets.

References

Barrett, L. F. (2016). The theory of constructed emotion: an active inference account of interoception and categorization. *Social Cognitive and Affective Neuroscience*, page nsw154.

Batbaatar, E., Li, M., and Ryu, K. H. (2019). Semantic-Emotion Neural Network for Emotion Recognition From Text. *IEEE Access*, 7:111866–111878.

Cowen, A. S. and Keltner, D. (2021). Semantic Space Theory: A Computational Approach to Emotion. *Trends in Cognitive Sciences*, 25(2):124–136.

Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., and Ravi, S. (2020). GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 4040–4054, Stroudsburg, PA, USA. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Drus, Z. and Khalid, H. (2019). Sentiment Analysis in Social Media and Its Application: Systematic Literature Review. *Procedia Computer Science*, 161:707–714.

Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200.

Pereira, D. A. (2021). A survey of sentiment analysis in the Portuguese language. *Artificial Intelligence Review*, 54(2):1087–1115.

Rosalind, P. (2000). *Affective Computing*. MIT Press, Cambridge.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: Pretrained BERT Models for Brazilian Portuguese. pages 403–417.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Learning rules for automatic identification of implicit aspects in Portuguese

Mateus Tarcinalli Machado[1], **Thiago Alexandre Salgueiro Pardo**[1], **Evandro Eduardo Seron Ruiz**[2], **Ariani Di Felippo**[3]

[1]Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo

mateusmachado@usp.br, taspardo@icmc.usp.br

[2]Faculdade de Filosofia, Ciência e Letras de Ribeirão Preto, Universidade de São Paulo

evandro@usp.br

[3]Núcleo Interinstitucional de Linguística Computacional (NILC)
Departamento de Letras, Universidade Federal de São Carlos

ariani@ufscar.br

Abstract. *This sentiment analysis work is focused on the task of identifying aspects, emphasizing the so-called implicit aspects, i.e., those that are not explicitly mentioned in the texts. For this, we analyzed frequency-based methods, adapted rules from the English language to Portuguese, and developed a method that learns new rules through corpus analysis.*

Resumo. *Este trabalho de análise de sentimentos está focado na tarefa de identificação de aspectos, dando ênfase aos chamados aspectos implícitos, ou seja, aqueles que não são mencionados explicitamente nos textos. Para isso, analisamos métodos baseados em frequência, adaptamos regras da língua inglesa para o português e desenvolvemos um método que aprende novas regras por meio de análise de corpus.*

1. Introduction

Sentiment analysis is an area of applied computing research related to natural language processing, which aims to analyze people's opinions, feelings, assessments, attitudes, and emotions concerning entities such as products, services, organizations, individuals, issues, events, topics, and their attributes [Liu 2012]. It presents a wide range of studies and is often referred to in the literature by slightly different names and tasks, such as opinion mining, opinion extraction and sentiment mining, among others. According to [Taboada 2016], sentiment analysis is a growing field at the intersection between linguistics and computer science that aims to determine automatic sentiments in a text.

Many authors [Medhat et al. 2014, Rana and Cheah 2016, Yadollahi et al. 2017] divide sentiment analysis into three levels: document, sentence, and aspect. Each level aims to address different challenges, with aspect-based analysis being the most refined. At the document level, the goal is usually related to the identification of the polarity of the whole document, to determine whether its overall opinion is positive or negative. This level of analysis is important in social and psychological studies carried

out on social networks, consumer satisfaction, analysis of patients in medical settings, and many others [Yadollahi et al. 2017]. Sentiment analysis performed at the sentence level aims to find the polarity of a sentence. At this level of analysis, it is also important to identify whether the target sentence is subjective or objective, i.e., whether it contains an opinion. This level of sentiment analysis is usually influenced by the context around the sentence and is considered very important for applications dealing with texts from social networks, comments and short messages, among others. Finally, the aspect level deals with more refined sentiment analysis strategies, trying to determine people's opinion about a specific characteristic (aspect) of a product, a service, or an entity [Medhat et al. 2014, Rana and Cheah 2016, Yadollahi et al. 2017]. To carry out this level of analysis, it is essential to identify the entities mentioned and their respective aspects and the so-called words of opinion related to the aspects. From there, the polarity of opinions directed to each identified aspect is determined. The results of the aspect-level sentiment analysis can then be finally summarized and visualized, for example, in a table containing the found aspects and their respective polarities [Rana and Cheah 2016]. This level of processing can also be called aspect-based sentiment analysis [Liu 2012] and is very useful for language processing applications, as text summarization [López Condori and Pardo 2017].

In this paper, we focus on the aspect detection task, emphasizing the treatment of implicit aspects. An aspect is considered explicit if it is mentioned directly in a sentence, otherwise, it is considered implicit [Liu 2012]. For example, in the sentence "The image quality of this camera is great", the expression "image quality" is an explicit aspect term. On the other hand, in "This camera is expensive", the word "expensive" implicitly refers to the "price" aspect.

It is a fact that we have a smaller amount of works that deal with implicit aspects [Ravi and Ravi 2015, Rana and Cheah 2016]. For Portuguese language, in particular, this is still a challenge [Pereira 2020]. Several authors [Zhang and Zhu 2013, Panchendrarajan et al. 2016] point out that between 15% and 30% of the aspects of opinionated texts are mentioned implicitly, and, for some categories, these percentages increase significantly. The non-detection of these aspects leads to the loss of important information and, consequently, negatively affects the results of related applications. The identification of implicit aspects and the treatment of texts in the Portuguese language, despite being complex problems, are relevant tasks and have the potential to advance the state of the art in the area.

This paper presents an analysis of frequency and rule-based methods for aspect detection. We also develop a method that learns extraction rules from a corpus. The rest of this work is organized as follows: Section 2 presents the main related work, especially the ones for Portuguese language. Section 3 presents our methods and datasets. Section 4 describes the results obtained by the implemented methods. Finally, Section 5 discusses the strengths and weaknesses of these methods.

2. Related work

There are few works that explicitly try to characterize, model and identify implicit aspects in the sentiment analysis area, which may be explained by the difficulties of dealing with them. We briefly synthesize the main related work in what follows.

The authors of [Cai et al. 2020] formulated the task as a problem of hierarchical prediction of categories and sentiments, where first the algorithm identify categories of aspects in a sentence and then the sentiments related to each detected category. To do this, they used a convolutional network of hierarchical graphs (Hier-GCN). To codify the sentences and their attributes, the authors used representations of bidirectional transformer encoders (BERT) [Devlin et al. 2018], which is a pre-trained model and has achieved excellent results in many natural language processing tasks. Their results show that this type of modeling is suitable for the detection of the aspect category together with the classification of sentiments, achieving in its highest result 0.76 precision, and 0.73 recall.

In [Marcacini et al. 2018], the authors developed a method based on heterogeneous networks that combines features as labeled aspects, unlabeled aspects, and linguistic features. To perform the classification, they developed an algorithm that propagates the labels using linguistic features as a bridge for this propagation. As a result, the method obtained f-measure between 0.56 and 0.68, depending on the analyzed domain.

The work of [Balage Filho 2017] analyzed the following aspect extraction methods: frequency-based, relation-based, and machine learning-based ones. The frequency-based method is the simplest. It selects the most frequent nouns and noun phrases as aspects. The relation-based one analyzes the relationships between aspects and the opinion related to them. In machine learning, an annotated training set is used to create a model that identifies the aspects. In their experiments, the frequency-based method showed the best result with 0.49 f-measure.

In the work of [Costa and Pardo 2020], the authors analyzed lexical-based methods for extracting aspects from opinions written in Portuguese. In a first experiment, the authors made use of aspect ontologies, in order to identify explicit and implicit aspects. In a second experiment, the authors extended a frequency detection method of aspects using the distributional models Word2Vec [Mikolov et al. 2013a] in order to enrich the process. The experiment performed with ontologies achieved the best results, with 0.53 precision, and 0.44 recall.

3. Data and Methods

3.1. Methods

In this work, we applied four methods for aspect detection: two frequency-based and two rule-based methods. For the frequency-based ones, we implemented the Freq Baseline and Freq Baseline with Word2Vec methods to serve as comparative bases for other methods. Regarding the rule-based methods, we tried to translate and apply the rules discovered in [Poria et al. 2014] to texts in Portuguese, as well as to use an automatic method for learning new detection rules. Before detailing the methods, we introduce the dataset that we use.

3.2. Dataset

For the execution of the experiments, we worked with a corpus formed by opinionated texts in Portuguese, resulting from the work of [Vargas and Pardo 2018]. The corpus is one of the few existent corpora with identified implicit aspects, even considering works in English. It has 60 product reviews from 3 different domains: cameras, books, and smartphones. Table 1 shows the characteristics of the corpus. It is important to note

that the percentages of implicit aspects found, with values close to 15% in the domains of cameras and smartphones, demonstrate the relevance of the specific analysis of this category of aspects. We evaluated all experiments in this corpus.

Table 1. Corpus composition.

Domains	Reviews	Aspects	Explicits	% Explicits	Implicits	% Implicits
Cameras	60	352	299	84.94%	53	15.06%
Books	60	330	304	92.12%	26	7.88%
Smartphones	60	455	387	85.05%	68	14.95%

In the corpus we found different cases of implicit aspects. For example, in "Very compact and very beautiful", the size and design aspects can be identified by their qualifiers. In "Camera of small measurement", the size aspect appears as a semantically close word. Finally, in "Works anywhere", only with specific knowledge about the context, in this case the functioning of the smartphone, we can understand that the phrase refers to the signal aspect. These are just a few examples, which demonstrate how complicated the task of identifying implicit aspects can be.

In a pre-processing stage, we prepared the texts for the execution of the methods. We selected these processes according to the needs of the algorithms. In our case, we performed the following processes:

- **Dataset division into training and test sets**: with the aim of consistently comparing the implemented methods, we performed all experiments using the same training (for searching parameters or rules, as we will see later) and testing (to validate the methods) datasets;
- **Sentences tokenization:** split of texts into sentences;
- **Words tokenization:** split of sentences into tokens, which can be words, symbols and punctuation, among others;
- **PoS tagging:** identification of morpho-syntactic categories of the tokens;
- **Dependency analysis:** construction of sentence dependency trees, to understand the relationship between tokens.

To perform these tasks, except for the division in training and testing sets, we used spaCy [Honnibal et al. 2020] module with pt_core_news_lg model from the Python programming language.

3.3. Method 1: Freq-Baseline

This method selects nouns and noun phrases as aspect candidates and analyzes their frequencies in relation to the number of sentences in the analyzed dataset. The algorithm selects the most frequent candidates as aspects. It is a method with an easy implementation that achieves good results, which is why it is often used as a basis for comparison to evaluate other algorithms.

The definition of the most frequent aspects is performed by comparing the frequency of the candidates being analyzed with a pre-defined cutoff frequency. In [Hu and Liu 2004], the authors used a cutoff frequency of 1%, a value that ended up being used in other studies as well. As in [Machado et al. 2017], in this work, we varied

this cutoff frequency and analyze what would be the most appropriate value. We worked with cutoff frequencies in an interval of 0.01% and 10% with an increment of 0.01% at each execution of the algorithm. Thus, at each iteration, we performed the calculations of precision, recall, f-measure, and percentages of explicit and implicit aspects correctly detected.

3.4. Method 2: Freq-Baseline + Word2Vec

As in [Pavlopoulos and Androutsopoulos 2014] and [Machado et al. 2017], in this work we used the distributional model Word2Vec [Mikolov et al. 2013b] to exclude candidates for aspects that are not related to the domain under analysis. The algorithm compared each candidate aspect with two vectors formed by the centroid of the Word2Vec vectors of words related to the context under study and of the Word2Vec vectors of general domain words. The candidate aspects closest to this general centroid vector were discarded, thus remaining only words more related to the context of the dataset. As it is a method used only as a baseline, we will not go into further details about it. More information can be found in the mentioned references. As in the previous experiment, we searched for the best cutoff frequencies, which remained the same.

3.5. Method 3: Adapted Rules of [Poria et al. 2014]

In our first experiment with rules, we adapted the rules from the work of [Poria et al. 2014]. The method comprises handcrafted rules that analyze the sentence dependency trees together with two lexicons: one with a list of implicit aspects and another called SenticNet [Cambria et al. 2014] that is a concept-level knowledge base containing a set of semantics, sentics, and polarities associated to natural language concepts.

As we did not find an implicit lexicon for the Portuguese language, we created one based on the existent corpus of [Vargas and Pardo 2018]. To maintain consistency with the other experiments and do not introduce bias in the results, we used only the terms found in the training set.

We translated the sentences of the presented examples in [Poria et al. 2014] and adapted the rules to Portuguese, changing the order or type of some components. Finally, we executed them in the adopted dataset.

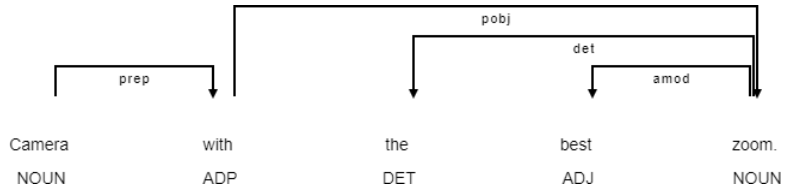
3.6. Method 4: Automatically Learned Rules

The results of the adaptations of the rules developed in [Poria et al. 2014] for Portuguese were below expectations. This motivated us to look for more adequate rules for the Portuguese language and with better detection of implicit aspects. For this, we used the dependency trees of the sentences of the training set.

We analyzed the elements that had some relationship of dependency on the aspects identified in the datasets and, together with the morpho-syntactic analysis performed, we created a set of rules. As an example, Figure 1 presents the dependency tree, obtained from spaCy, of the sentence “Camera with the best zoom.”. In this sentence, the term “zoom” is marked as an aspect, therefore, for the formation of the rules, the terms that have some relationship with it were used.

In the first stage, the elements that have a direct relationship with the marked aspects were selected. The relations and morpho-syntactic characteristics of the elements

Figure 1. Example of dependency tree.



were combined, forming aspect extraction rules. For example, see the rules presented in Table 2. We can interpret the first rule of this table as follows: if a noun (NOUN) is in an adjectival modifier relationship (amod) with an adjective (ADJ), then the first token is extracted as aspect. This same process was repeated, but this time analyzing combinations of three, four, and five tokens, thus generating 1,200 rules for camera domain, 1,470 for books, and 1,564 for smartphones.

Table 2. Example of learned rules.

Tokens		Rules			
Token 1	Token 2	Class 1	Relation	Class 2	Aspect
zoom	best	NOUN	amod	ADJ	Token 1
		*	amod	ADJ	Token 1
		NOUN	*	ADJ	Token 1
		NOUN	amod	*	Token 1
		NOUN	*	*	Token 1
		*	amod	*	Token 1
		*	*	ADJ	Token 1
with	zoom	ADP	pobj	NOUN	Token 2
			⋮		

After the execution of the rule generation process with all the sentences in the corpus, the next step was the selection of the best rules. For that, the rules were created using the class DependencyMatcher of the module spaCy in the language Python, and each one was executed in the training set.

We calculated, for each rule, the Laplace precision and support metrics. We tested rules with a support value between 1 and 60 combined with a Laplace precision ranging between 0.1 and 1, with an increment of 0.1. For each combination of Laplace precision and support, we selected the rules that met these requirements, ran them on the test suite, and calculated the precision, recall, and f-measure metrics. To select the best set of rules, we choose the rules that got the highest result for f-measure.

4. Results

Our first experiments should serve as baseline results. Although simple, we already have a first contribution with the analysis of cutoff frequencies. As mentioned, several authors use the value of 1% as a cutoff. We find a cutoff frequency between 0.40% and 0.53% as a more appropriate value. Using the pruning mechanism with Word2Vec, as in other works, led to an increase in accuracy and a decrease in recall. The identification of implicit aspects, as expected because of the characteristics of the method, was practically non-existent. The results are summarized in Table 3.

Table 3. Results for Frequency-based methods.

Domain	Method	% Cutoff	Precision	Recall	F-measure	% Explicits	% Implicits
Camera	Freq	0.53	0.57	0.57	0.57	35.16	0.00
Camera	Freq+W2V	0.53	0.72	0.56	0.63	34.07	0.00
Book	Freq	0.42	0.47	0.67	0.55	23.17	0.00
Book	Freq+W2V	0.42	0.70	0.59	0.64	15.85	0.00
Smartphone	Freq	0.40	0.46	0.53	0.49	27.61	4.17
Smartphone	Freq+W2V	0.40	0.56	0.47	0.51	23.88	0.00

As mentioned, the results of the experiment with the rules adapted from [Poria et al. 2014] did not reach satisfactory results. Therefore, we focused on finding new rules that would be more appropriate for the Portuguese language. The results are summarized in Table 4. For the camera domain, we got the best f-measure result with 26 rules selected by a minimum Laplace precision of 0.30 and support of 6. Regarding the book domain, we only found 1 rule selected by the Laplace precision of 0.40 and support of 21. Finally, in the smartphone domain, we found 3 rules selected by the Laplace precision of 0.10 and support of 18.

Table 4. Results for learned rules.

Domain	Laplace[1]	Support	Rules	Precision	Recall	F-measure	% Explicits	% Implicits
Camera	0.30	6	26	0.50	0.58	0.54	36.26	15.38
Book	0.40	21	1	0.81	0.54	0.65	13.41	20.00
Smartphone	0.10	18	3	0.61	0.42	0.50	20.90	4.17

Analyzing the learned rules, we could observe that most of the rules show nouns as aspects, changing only the classes and relationships with other terms.

We also selected the common rules between the domains, to find rules that were theoretically less susceptible to the context. Table 5 shows the rules that were found, and, as we can see, there was a greater number of rules found between the domains of cameras and smartphones because of the greater similarity between them. The common rules between smartphone and book were the same as found between camera, smartphone, and book. The first rule can be interpreted as: any token in a nominal subject relation with a noun that is in a determiner relation with a determiner.

Table 5. Common rules between domains.

Sets	Rule	Aspect
camera ∩ smartphone	['NOUN', 'nsubj'], '>', ['DET', 'det']	Token 1
	['NOUN', 'obj'], '>', ['DET', 'det']	Token 1
	['NOUN', 'conj'], '>', ['PUNCT', 'punct']	Token 1
camera ∩ smartphone ∩ book	['NOUN', 'nsubj'], '>', ['DET', 'det']	Token 1

The results were inferior to the previous experiment, as we can see in Table 6. There was no improvement, even in terms of accuracy, that could have increased with a smaller number of rules. This result was probably due to the simplicity of the found rules, which, as previously mentioned, always ended up selecting only nouns as aspects, similar to what occurs in the frequency-based methods.

Table 6. Results for rules common to all the domains.

Domain	Rules	Precision	Recall	F-measure	% Explicits	% Implicits
Camera	Camera ∩ Smartphone	0.49	0.31	0.38	15.38	0.00
Camera	Camera ∩ Smartphone ∩ Book	0.62	0.12	0.21	5.49	0.00
Book	Camera ∩ Smartphone	0.53	0.57	0.55	17.07	20.00
Book	Camera ∩ Smartphone ∩ Book	0.81	0.54	0.65	13.41	20.00
Smartphone	Camera ∩ Smartphone	0.61	0.42	0.50	20.09	4.17
Smartphone	Camera ∩ Smartphone ∩ Book	0.73	0.24	0.36	10.45	0.00

Finally, Table 7 presents a summary of the results got by each method. We can observe that only the rules we found could find implicit aspects in all domains. Considering f-measure, in the camera domain, the frequency-based method with Word2Vec was superior and, in the other domains, it was practically equivalent to the results of the rules, although it was not able to find implicit aspects.

Table 7. Results by implemented method.

	Camera		Book		Smartphone	
Method	F-measure	% Implicits	F-measure	% Implicits	F-measure	% Implicits
Freq	0.57	0.00	0.55	0.00	0.49	4.17
Freq + Word2Vec	**0.63**	0.00	0.64	0.00	**0.51**	0.00
Poria et al. rule-based	0.10	0.00	0.14	0.00	0.14	**7.35**
Corpus-based rules	0.54	**15.38**	**0.65**	**20.00**	0.50	4.17

5. Conclusion

Regarding the frequency-based methods, we could observe that, despite being simple, they achieved interesting results. With the study of the cutoff frequency, we could still get a significant improvement in these results. Using Word2Vec also proved to be effective in eliminating some aspects that were wrongly identified, a fact observed by the increase in precision. On the other hand, it ended up eliminating the few implicit aspects that had been identified.

The adaptation of the rules from English to Portuguese proved inefficient. Although at first the analysis of the examples of the rules proved promising, in the experiments, the results were below expectations. The method for finding rules in our last experiment has shown promise. Although we did not get a vast improvement in the f-measure, the improvement in implicit aspect detection was substantial. In future work, we intend to consider additional elements for analysis and creation of rules, such as sentiment lexicons, and to separately search for rules for explicit and implicit aspects.

The interested reader may find more information at the web portal of the POeTiSA (*POrtuguese processing - Towards Syntactic Analysis and parsing*) project[2].

Acknowledgements

The authors are grateful to the Center for Artificial Intelligence (C4AI), with support of the São Paulo Research Foundation (grant #2019/07665-4) and IBM Corporation.

[2] https://sites.google.com/icmc.usp.br/poetisa

References

Balage Filho, P. P. (2017). *Aspect extraction in sentiment analysis for portuguese language*. PhD thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brazil.

Cai, H., Tu, Y., Zhou, X., Yu, J., and Xia, R. (2020). Aspect-category based sentiment analysis with hierarchical graph convolutional network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 833–843, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Cambria, E., Olsher, D., and Rajagopal, D. (2014). Senticnet 3: a common and commonsense knowledge base for cognition-driven sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Costa, R. W. M. and Pardo, T. A. S. (2020). Métodos baseados em léxico para extração de aspectos de opiniões em português. In *Anais do IX Brazilian Workshop on Social Network Analysis and Mining*, pages 61–72. SBC.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

López Condori, R. E. and Pardo, T. A. S. (2017). Opinion summarization methods: Comparing and extending extractive and abstractive approaches. *Expert Systems with Applications*, 78:124–134.

Machado, M. T., Pardo, T. A. S., and Ruiz, E. E. S. (2017). Analysis of unsupervised aspect term identification methods for portuguese reviews. *Anais do XIV Encontro Nacional de Inteligência Artificial e Computacional (ENIAC), SBC*, pages 239–249.

Marcacini, R. M., Rossi, R. G., Matsuno, I. P., and Rezende, S. O. (2018). Cross-domain aspect extraction for sentiment analysis: A transductive learning approach. *Decision Support Systems*, 114:70–80.

Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

Panchendrarajan, R., Ahamed, N., Murugaiah, B., Sivakumar, P., Ranathunga, S., and Pemasiri, A. (2016). Implicit aspect detection in restaurant reviews using cooccurence of words. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 128–136.

Pavlopoulos, J. and Androutsopoulos, I. (2014). Aspect Term Extraction for Sentiment Analysis: New Datasets, New Evaluation Measures and an Improved Unsupervised Method. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)@ EACL*, pages 44–52.

Pereira, D. A. (2020). A survey of sentiment analysis in the portuguese language. *Artificial Intelligence Review*, pages 1–29.

Poria, S., Cambria, E., Ku, L.-W., Senticnet, C. G., and Gelbukh, A. (2014). A Rule-Based Approach to Aspect Extraction from Product Reviews. In *Proceedings of the second workshop on natural language processing for social media (SocialNLP)*, pages 28–37.

Rana, T. A. and Cheah, Y.-N. (2016). Aspect extraction in sentiment analysis: comparative analysis and survey. *Artificial Intelligence Review*, 46(4):459–483.

Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems*, 89:14 – 46.

Taboada, M. (2016). Sentiment analysis: An overview from linguistics. *Annual Review of Linguistics*, 2:325–347.

Vargas, F. A. and Pardo, T. A. S. (2018). Aspect Clustering Methods for Sentiment Analysis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11122 LNAI:365–374.

Yadollahi, A., Shahraki, A. G., and Zaiane, O. R. (2017). Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)*, 50(2):1–33.

Zhang, Y. and Zhu, W. (2013). Extracting implicit features in online customer reviews for opinion mining. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 103–104.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Text Mining for Cyberbullying Detection: a Brazilian Portuguese Evaluation

Carolina Eberhart[1], **Luciano Ignaczak**[1], **Márcio Garcia Martins**[1]

[1] Universidade do Vale do Rio do Sinos (UNISINOS) – São Leopoldo, RS – Brasil

carolieberhart@gmail.com, lignaczak@unisinos.br, marciog@unisinos.br

Abstract. *Bullying and cyberbullying are words commonly seen in today's news. Although the scientific community has evaluated text mining techniques for cyberbullying detection, few studies have targeted Brazilian Portuguese datasets. Our study aims to assess the text mining application to detect cyberbullying messages written in Brazilian Portuguese. We gathered posts and comments from Reddit communities and extracted several text features. We then processed these features using Naïve Bayes and SVM classifiers to uncover cyberbullying activity. The outcomes of this experiment may not be used solo for cyberbullying detection; however, they can aid moderators in prioritizing content reviews and acting faster on real cyberbullying cases.*

Resumo. *Bullying e cyberbullying são assuntos abordados com frequência pela mídia. Embora a comunidade científica venha avaliando técnicas de mineração de texto para detecção de cyberbullying, poucos estudos utilizam datasets em português. Este estudo tem como objetivo avaliar a aplicação de mineração de texto para detectar mensagens em português associadas com cyberbullying. O estudo coletou posts e comentários de comunidades do site Reddit e extraiu diversas features, que foram usadas para treinar classificadores para descoberta de cyberbullying. Apesar dos resultados não demonstrarem que mineração de texto possa automatizar completamente a detecção de cyberbullying, as técnicas podem auxiliar moderadores na priorização da análise de mensagens.*

1. Introduction

Bullying and cyberbullying are not recent phenomenon. [Smith et al. 1999] defined bullying as "a subcategory of aggressive behavior; but a particularly vicious kind of aggressive behavior, since it is directed, often repeatedly, towards a particular victim who is unable to defend himself or herself effectively". [Hinduja and Patchin 2014] expanded this definition by stating that cyberbullying is an extension of bullying that includes harassment through electronic devices. [McCarthy 2018] gives a notion of the relevance of this problem when he mentions that, in 2018, 37% of Indian parents reported that their children experienced cyberbullying. Also, over 20% of the parents in the United States, Brazil, South Africa, and Canada reported the same issue. Still, cyberbullies target people of all ages and backgrounds.

The necessity of automated detection of cyberbullying messages on the web is unquestionable. Text mining provides a way of automating the message filtering process, and machine learning can offer support to the process. According to [Taeho 2019], text mining is the process of extracting knowledge from textual data. The usage of

text mining to inhibit cyberbullying activities also shows in the scientific community worldwide. [Zhao and Mao 2016] study focuses on using text analysis to identify if English messages on Twitter are cyberbullying, even if those do not contain insults. [Nandhini and Sheeba 2015a]'s work focuses on analyzing English posts from Formspring and Myspace, categorizing those into bullying types such as harassment or racism. [Urtiga and Castro 2018] study uses data from Brazilian Portuguese Twitter messages to determine message topics that usually indicate bullying or cyberbullying activity. Studies utilized different text mining tasks and classification models to achieve their goals. However, a significant number of them incorporate an insulting words dictionary combined with another technique that assists in getting the context of the messages.

Our study aims to evaluate a text mining proposal for cyberbullying detection in messages written in Brazilian Portuguese. To achieve this, we collected posts and comments from specific Reddit communities and extracted information such as the text sentiment, key phrases, and the presence of insults. We then ran these features through two text classification algorithms: Naïve Bayes and SVM. Then, we also assessed how text mining could assist in finding the most toxic Reddit communities and users by ranking them based on our results. We want to contribute to the future state of automatic moderation of Brazilian Portuguese language online communities specifically. Although there is academic work on this subject, we only found a small number of studies focused on Brazilian Portuguese datasets.

This paper is organized as follows. In Section 2, we discuss recently published scientific papers that concern using text mining tasks in cyberbullying identification. Section 3 describes our research methodology, including the data source, chosen features, and the classifiers. In Section 4, we discuss the results obtained from the execution of our experiment and point out its contributions. Finally, Section 4 presents the conclusions and proposals for future work.

2. Related work

This section was based on using the search string "cyberbullying AND text mining" to find papers published on Association for Computing Machinery (ACM), Institute of Electrical and Electronics Engineers (IEEE Xplore), and Science Direct bases. The search string returned 72 papers. Then, we filtered the list by eliminating any papers published before 2015 or that had a Google H5-Index value lower than 15. The remaining ones were manually evaluated by their abstract contents, leaving a total of seven papers. Meanwhile, the second string "bullying AND mineração de dados" was used in Google Scholar to find related works specifically in Brazilian Portuguese. The strings resulted in 81 papers, which were filtered using the same published year and Google H5-Index criteria. Still, as Google H5-Index does not cover some of the smaller Brazilian journals, we also filtered the results by eliminating any papers with a Qualis "C" evaluation. Then, after reading the remaining results' abstracts, only one paper was deemed similar to our work. Therefore, this section presents the eight related works.

Every paper selected conveys a variation of cyberbullying identification techniques, providing a text-processing methodology combined with classification or clusterization tasks. These papers' general focus is building a classification or clusterization model for either binary message classification, non-binary message classification, or cy-

berbullying risk discovery.

The related papers under the binary message classification category focused on analyzing web content and evaluating whether it is cyberbullying or not. They all track the presence of insults in the messages; however, they do not consider this as the only significant feature to be analyzed. [Zhao et al. 2016] proposed evaluating the semantics and linguistic relationships of the words in the messages, as these also hold great importance in detecting cyberbullying. [Zhao and Mao 2016] also proposed using semantic and linguistic evaluation; however, their work focuses on creating a model to identify cyberbullying in messages that do not contain insults. [Singh et al. 2016] had a distinct approach to the problem. These authors' methodology was based on text and social features, such as the number of connections and their degree of centrality in the social network.

In the second category of papers - non-binary message classification - cyberbullying messages were classified into different cyberbullying categories (e.g., flaming, harassment, racism). To achieve this, [Nandhini and Sheeba 2015a] used the presence of insulting words, word frequencies, and part-of-speech as features. In another paper, [Nandhini and Sheeba 2015b] used fuzzy rules and a genetic algorithm to create their model, which was also based on part-of-speech and word frequency features.

The papers categorized as cyberbullying risk discovery aimed to identify words and expressions that indicate the presence or the risk of future cyberbullying activity occurrences and the topics associated with these cyberbullying messages. [Song and Song 2020] used features such as words related to cyberbullying methods and causes and the overall message sentiment. The final classification presents the level of cyberbullying risk associated with each message. Meanwhile, [Song et al. 2020] classified the messages into predetermined risk categories, using features such as term and document frequency, degree of diffusion, and degree of visibility. Finally, the study of [Urtiga and Castro 2018] was based on messages that already portray the cyberbullying theme but are not necessarily cyberbullying. They perform text clustering utilizing word frequencies as a feature.

Like most of the aforementioned papers, this work evaluated the automatic detection of cyberbullying activity. As in seven of the analyzed papers, we use text classification algorithms to detect cyberbullying. Furthermore, like [Singh et al. 2016], we utilize multiple classification algorithms and compare their results. Also, our feature selection focuses on sentiment analysis - as in [Song and Song 2020] - and the detection of insulting words - as in [Zhao et al. 2016, Zhao and Mao 2016, Nandhini and Sheeba 2015a]. The main distinction is that we search for the insults in the whole documents and also on their key phrases. We created this division to determine if the insults are a crucial part of the document or not. Finally, our dataset for this study was in Brazilian Portuguese.

3. Research Methodology

Our evaluation for automatic identification of cyberbullying messages along with harmful communities and users explored the Reddit (https://www.reddit.com/) website using a group of Python scripts. We chose Python as a programming language because of its text mining and Reddit information extraction libraries. The following sections expose the details regarding this study's dataset source and language, feature selection methods, and text classification algorithms. The scripts created for this experiment, as well as the

utilized data and obtained results, are available on our GitHub[1].

3.1. Data source and collection

Reddit is a very active website, and it hosts thousands[2] of communities in many languages, including Brazilian Portuguese, which is our specific target in this study. Since Reddit does not have a list of subreddits by language, we used the emportugues.org website (https://emportugues.org/), which contains a list of Portuguese language subreddits, to obtain the base of our data source. On April 11th, 2021, we collected the complete list of 1993 communities' data from the emportugues.org website. We then filtered the records which would not be valuable for our study: duplicated, non-Brazilian Portuguese, and less active - less than 1,000 members - communities. The final dataset obtained from this filtering process contained 133 communities.

Then, on April 12th, 2021, we collected the actual data used in our experiment - the community comments - by utilizing the PRAW[3] (Python Reddit API Wrapper) library to access the selected communities' top 10 topics from the past 12 months - April 13th, 2020 to April 12th, 2021. We then cleared out each comment's emojis before storing them in our database, along with the respective author and community name. Through this process, we were able to obtain a total of 30,634 comments.

3.2. Data preprocessing

To obtain an appropriate dataset for text mining, we performed a few preprocessing steps. We began by removing comments that were too small in length to provide valuable information. According to [Song et al. 2014], the problem of short text classification presents itself differently from the general text classification one. Short text tends to be sparse, containing only a few words, few features, and a low level of information that does not provide context. [Song et al. 2014] mentions 30 characters news titles as short text examples, so we used this number as the base for our experiment and eliminated comments with less than 30 characters.

We also observed the presence of blank, foreign language, and bot comments in our data. Since we are interested in identifying cyberbullying activity as well as toxic users and communities in Brazilian Portuguese, none of these comments would be valuable in our dataset. Therefore, we removed all of their instances from our data. Our final dataset contained a total of 19,272 comments, which we then manually labeled as cyberbullying or not cyberbullying.

3.3. Selected features

We selected 12 features for our experiment, some based on related work, some in the characteristics of cyberbullying, and others in our knowledge of the internet forum Reddit. We then categorized these features into three groups: document sentiment features, insult / explicit wording presence features, and URL presence features.

The first set of features - positive, neutral, and negative sentiment confidence of the overall document - is meant to identify the possibility of maliciousness in a comment.

[1] https://github.com/ceberhart2611/Cyberbullying-article-2021

[2] https://www.redditinc.com/press

[3] https://praw.readthedocs.io/en/latest/

We based this set of features on the fact that insults in a message may not always mean it is malicious. Therefore, the sentiment would assist in deciding the message tone and aid the classifier in a decision not exclusively based on certain words. We obtained these sentiment confidence features by utilizing the Microsoft Azure Text Analysis API sentiment analysis module[4]. This API module mines the text for positive, neutral, and negative sentiment clues and outputs a confidence score between zero and one for each sentiment. Thus, the sum of these positive, neutral, and negative sentiment confidences for a given text always equals one.

The second set of features - the number of insults and sexually explicit words in the document and its key phrases - adds the possibility of explicitly identifying aggression and sexual harassment. We based this set of features on the articles we found in our related work search - [Zhao et al. 2016]; [Singh et al. 2016]; [Nandhini and Sheeba 2015a]; [Choi et al. 2020]. However, in all of these articles, a single dictionary of insulting words is used. Thus, we implemented a different approach by assigning insults into four dictionaries according to their category, allowing a broader set of insults in our experiment and enabling the classifier to work with different aggression types separately.

We created three different dictionaries of insulting words and one sexually explicit language dictionary containing adjectives and nouns only. The first insulting words dictionary contains only swear words, the second general insults, and the third context-related insults. General insults are offensive but socially accepted as they do not present inappropriate terminology. ”Shit” and ”dumbfuck” are examples of swear words, while ”idiot” and ”stupid” are general insults. Meanwhile, context-related insulting words are the ones that not only do not work solo as insults but also carry generally unoffensive meanings. For instance, saying ”I have pigs on my farm” is not insulting; however, saying ”Jack is such a pig” carries an offensive tone. These dictionaries aim to find the general cases of aggression tied to cyberbullying activity while still allowing other features to provide context cues. For example, even though swear words are generally unacceptable, they still can be used without offensive intent.

The sexually explicit words dictionary has the specific goal of assisting in revealing instances of sexual harassment. Its contents include jargon for body parts, sexual positions, and sexual acts. However, even though anatomic words such as ”penis” or ”vagina” may also indicate sexual harassment, these were not included in the dictionary. Reddit hosts many communities where healthy discussions about sexuality and sexual relationships, and these terms may be used in those. The final set, including the three insulting words dictionaries and the sexually explicit terms, has 1,191 unique records. The complete dictionaries[5] and sexually explicit words[6] used are available on GitHub.

In general, these features are build by finding the number of matches between each dictionary and a text record, resulting in two features per dictionary - one for the

[4] https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/text-analytics-how-to-sentiment-analysis?tabs=version-3-1

[5] https://github.com/ceberhart2611/Cyberbullying-article-2021/blob/main/insult-dictionaries.csv

[6] https://github.com/ceberhart2611/Cyberbullying-article-2021/blob/main/sex-slang-dictionary.csv

comment's whole text and then one for this comment's key phrases. We extracted the key phrases of each document by utilizing the Microsoft Azure Text Analysis API's key phrases module[7], which outputs the main points of a given unstructured text document. The goal of using key phrases instead of only the complete document to obtain features is to check if being aggressive or offensive is a core part of a comment.

Finally, the third group is a single feature: the presence of URLs in a comment. This feature indicates if a URL is present in a comment by trying to find "HTTP://", "HTTPS://", or "www." within its text. While some websites generate direct URLs to their content using content IDs, Reddit creates links utilizing the name of its users and communities or the title of the posts. Thus, a user posting a URL for a community or a user profile with an insulting name, or a post with an insulting title, may get unintended matches in sexually explicit and insulting words features. The general goal of tracking the URL presence is to give a chance for our classifiers to weigh these specific cases differently. This feature is also a contribution of this article as we did not find it in our related work corpus.

3.4. Data partitioning and classifiers

As our experiment utilizes supervised classifiers, we split our data into two different training and testing sets. 90% training / 10% testing and 80% training / 20% testing. Still, we identified that our testing results might not be satisfactory as, even though we had a high percentage of training cases, our dataset is highly imbalanced. In a total of 19,272 records, only 17.6% of them are labeled as cyberbullying. To mitigate this data imbalance issue, we used the Synthetic Minority Oversampling Technique - also known as SMOTE - as proposed by [Chawla et al. 2002]. The application of this oversampling method resulted in balanced training datasets, which had 50% of cyberbullying records and 50% of non-cyberbullying records. The specific SMOTE code used in this experiment is part of version 0.8.0 of the imblearn[8] Python library.

To acquire a fair evaluation of the efficiency of our features, we also used two distinct classification algorithms: Naïve Bayes and SVM. The choice of Naïve Bayes and SVM algorithms was associated with their significant presence in the related work and their different approaches. While Naïve Bayes does not explore relationships between the features and assumes their independence, SVM explores the possible relationships between them. The goal of processing our feature set in different classifiers was to verify how their different approaches perform on our dataset. As discussing classifiers implementation and optimization was not part of our scope, we used the Gaussian Naïve Bayes and the linear SVM implementations available on SciKit-learn.

4. Results and discussion

In this section, we discuss the results obtained by applying our research methodology. Table 1 shows that, through our evaluation, we obtained accuracy as high as 81%, and in general, the variations of our performance metrics do not surpass three percentage points. This same table also shows that our highest values for accuracy and precision are

[7] https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/text-analytics-how-to-keyword-extraction

[8] https://imbalanced-learn.org/stable/

81% and 45%, which appear under both the Naïve Bayes 80/20 column and the SVM 80/20 column. Still, the highest recall - our primary metric - and F1-score are under the SVM 80/20 column, which portrays the results of processing our features using an SVM classifier with an 80% training, 20% testing data fold. Hence, we concluded that this approach, which we will call SVM8020 onwards for simplicity, is the best result of our experiment.

Tabela 1. Experiment results - performance metrics

	Naïve Bayes		**SVM**	
Metric / Distribution	**90/10**	**80/20**	**90/10**	**80/20**
Accuracy	0.80	**0.81**	0.79	**0.81**
Precision	0.43	**0.45**	0.42	**0.45**
Recall	0.53	0.54	0.53	**0.56**
F1-score	0.47	0.49	0.47	**0.50**

Since presenting the results related to the top 10 toxic users and Reddit communities for our four different approaches would result in an extensive set of tables, we opted to show only the ones related to SVM8020. This evaluation was the one that obtained the best recall, meaning that it captured most of the comments deemed cyberbullying correctly. We had, however, to perform some manipulation of the data presented in the testing records before we obtained our final ranking. Therefore, communities and users with less than 14 and five comments were eliminated from the ranking data to allow a fair evaluation of their behavior.

Table 2 shows the outcomes of this process, comparing user and community toxicity as obtained from the actual comment labels and the predicted ones. The results shared in the table reveal that SVM8020 pointed at least half of the most toxic users and communities correctly; however, it was utterly deficient in the ranking order. Given that SVM8020 achieved a recall value of 56%, we can state that this result is satisfactory. However, if the prediction can only find around half of the actual cyberbullying cases, it will not correctly order the top offending authors and communities.

In order to find the issues in our predictions, we had to explore the misclassified records. When looking into the false negatives, we were able to identify cases in which the insults were not in our dictionary or were in the verbal form, which we did not explore. Commonly, the cases identified as cyberbullying in our dataset involved using some insults combined with a generally negative sentiment. Hence, cyberbullying cases in which there were no insulting words or that contained some insults, but had a highly neutral sentiment, were also not caught. Furthermore, we observed cases in which insults were combined with other words - for instance, ”shitface” - which made them go unnoticed as we searched for exact matches. Finally, we found that internet slang and instances of sarcasm also impacted our predictions and generated false negatives.

Meanwhile, in the false positives group, we found many insulting comments unrelated to people or groups, but rather companies, products, or the comment author himself. We also found instances of insulting words being used to intensify the effect of praising something, for instance, saying, ”fucking amazing”. Finally, the last common false-positive cases are tied to phrases with a high percentage of negative sentiment con-

Tabela 2. Most toxic users and communities in percentage according to the actual comments labels and the SVM8020 label predictions

	Actual label		Predicted label	
Rank	**Toxic users**	**Toxic communities**	**Toxic users**	**Toxic communities**
1	user560*	comm23*	user951*	comm5
2	user1776	comm107*	user560*	comm40*
3	user1302*	comm40*	user2722	comm51
4	user2528*	comm84*	user1302*	comm107*
5	user411	comm43	user2528*	comm53
6	user951*	comm86	user433	comm23*
7	user1747	comm45	user1094	comm104
8	user388	comm35*	user84	comm41
9	user1626	comm26	user839	comm35*
1	user1962	comm13	user2207	comm84*

* Users or communities ranked in the top ten for toxicity both by the actual labels and the predicted labels

fidence that contain only context-related insults. In these cases, the sentiment is generally sadness, not maliciousness; however, our sentiment analysis does not distinguish types of negativity.

Given all the performance metrics and the analysis we shared about our experiment, it is fair to say that the overall results obtained are satisfactory and may be applied to real-world scenarios of cyberbullying identification. Even though 56% of recall would not allow our automatic cyberbullying identification approach to work solo on Reddit, it could help the website moderation team to prioritize the investigation of specific comments and act faster when it comes to real cyberbullying cases. However, we would not recommend using the most toxic users and communities raking, as the ordering piece could misguide the exploration of the source of the most harmful content on the website.

Conclusion

In this article, we evaluated text mining for the identification of cyberbullying messages written in Brazilian Portuguese. Our evaluation comprised the presence of insults and the sentiment of text comments in Reddit communities. We verified the performance by running our dataset through two different text classifiers and then measuring it using accuracy, precision, recall, and F1-score metrics. The results of our experiment were overall satisfactory, yielding the possibility of its application in real-world scenarios. Therefore, we determined that text mining is helpful for the automatic moderation of Brazilian Portuguese online communities. In conclusion, this work sets a starting point for exploring machine learning-based online moderation in the Brazilian Portuguese language.

As the next step for future research, we suggest adding entity recognition and insulting verbs to our features to improve the recall percentage in the experiment. We also suggest considering moving the sexual harassment identification to another research more specifically tied to this subject and involving image processing techniques. Even though sexual harassment also pertains to the cyberbullying subject, it contains particularities that could benefit from joining text mining and image analysis.

Referências

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Choi, Y.-J., Jeon, B.-J., and Kim, H.-W. (2020). Identification of key cyberbullies: A text mining and social network analysis approach. *Telematics and Informatics*, page 101504.

Hinduja, S. and Patchin, J. W. (2014). *Bullying beyond the schoolyard: Preventing and responding to cyberbullying*. Corwin Press.

McCarthy, N. (2018). Where cyberbullying is most prevalent. Statista, 2018. Available at: <https://www.statista.com/chart/15926/the-share-of-parents-who-say-their-child-has-experienced-cyberbullying/>. Acessed in: November 24, 2020.

Nandhini, B. S. and Sheeba, J. (2015a). Cyberbullying detection and classification using information retrieval algorithm. In *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology*, pages 1–5.

Nandhini, B. S. and Sheeba, J. (2015b). Online social network bullying detection using intelligence techniques. *Procedia Computer Science*, 45:485–492.

Singh, V. K., Huang, Q., and Atrey, P. K. (2016). Cyberbullying detection using probabilistic socio-textual information fusion. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 884–887. IEEE.

Smith, P. K., Catalano, R., Junger-Tas, J., Slee, P., Morita, Y., and Olweus, D. (1999). *The nature of school bullying: A cross-national perspective*. Psychology Press.

Song, G., Ye, Y., Du, X., Huang, X., and Bie, S. (2014). Short text classification: A survey. *Journal of multimedia*, 9(5):635.

Song, J., Han, Y., Kim, K., and Song, T. M. (2020). Social big data analysis of future signals for bullying in south korea: Application of general strain theory. *Telematics and Informatics*, 54:101472.

Song, T.-M. and Song, J. (2020). Prediction of risk factors of cyberbullying-related words in korea: Application of data mining using social big data. *Telematics and Informatics*.

Taeho, J. (2019). Text mining concepts, implementation, and big data challange,(p. 1). *Seoul, Korea: Hongik University*.

Urtiga, T. and Castro, T. (2018). Detecção de bullying escolar em redes sociais e suas implicações na educação de adolescentes. In *Brazilian Symposium on Computers in Education (SBIE)*, volume 29, page 1693.

Zhao, R. and Mao, K. (2016). Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoder. *IEEE Transactions on Affective Computing*, 8(3):328–339.

Zhao, R., Zhou, A., and Mao, K. (2016). Automatic detection of cyberbullying on social networks based on bullying features. In *Proceedings of the 17th international conference on distributed computing and networking*, pages 1–6.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Relation extraction in structured and unstructured data: a comparative investigation on smartphone titles in the e-commerce domain

João Gabriel Melo Barbirato[1], **Livy Real**[2], **Helena de Medeiros Caseli**[1]

[1] Federal University of São Carlos – Computing Department
Caixa Postal 676 – 13565-905 – São Carlos – SP – Brazil

[2]Digital Lab – americanas s.a.
São Paulo SP, Brazil

jgmbarbirato@estudante.ufscar.br, helenacaseli@ufscar.br

livy.coelho@b2wdigital.com

Abstract. *As large amounts of unstructured data are generated on a regular basis, expressing or storing knowledge in a way that is useful remains a challenge. In this context, Relation Extraction (RE) is the task of automatically identifying relationships in unstructured textual data. Thus, we investigated the relation extraction on unstructured e-commerce data from the smartphone domain, using a BERT model fine-tuned for this task. We conducted two experiments to acknowledge how much relational information it is possible to extract from product sheets (structured data) and product titles (unstructured data), and a third experiment to compare both. Analysis shows that extracting relations within a title can retrieve correct relations that are not evident on the related sheet.*

1. Introduction

The main purpose of extracting information from text is to transform it into useful and well-structured knowledge [Pawar et al. 2017]. This can be done by means of well-known Natural Language Processing (NLP) tasks such as named-entity recognition, Information Extraction (IE) or Relation Extraction (RE).

Relation Extraction consists in automatically identifying relations in unstructured textual data [Pawar et al. 2017]. In the general domain, relationships instantiate facts with a high probability of being true (or highly plausible) [Xu et al. 2020]. But relation extraction in specific domains is also challenging, due to factors such as the higher variability of vocabulary, noisy and missing data, and the lack of standardization that is common in real scenarios. To exemplify this, next we show three product titles, in Portuguese, found on americanas.com:[1]

S1 `smartphone multilaser ms40s preto 4" câmera 3 mp + 5 mp 3g quad core 8gb android 6.0 p9025`
S2 `smartphone samsung galaxy s5 sm g900m branco tela 5.1", android 4.4, 4g, câmera 16mp`
S3 `celular positivo 2.4" 3g bluetooth fm mp3 p30c preto`

[1]Extracted in November 2020.

These are three different products in the smartphone (and cellphone) category. From these examples it is possible to identify three different brands (`multilaser`, `samsung` and `positivo`), two colors (`preto` and `branco`), two versions of the operating system Android (`6.0` and `4.4`) and two different camera resolutions (`3 mp + 5 mp` and `16 mp`). These are examples of product properties that could give rise to binary relations with the item being offered (the smartphone or cellphone).

In this context, this work aims to investigate how relations that were automatically extracted from unstructured data using BERT [Devlin et al. 2019] can enhance the information extracted from structured data. Bidirectional Encoder Representations from Transformers (BERT) is an encoder architecture capable of applying transfer learning for downstream NLP tasks through the fine-tuning process [Devlin et al. 2019]. In [Soares et al. 2019], the authors show that the encoder can also be used for RE from a corpus annotated with relations of interest. Thus, in this paper we present some experiments carried out with BERT Relation Extraction[2] to extract binary relations from e-commerce data.

The main contributions of this work are: (i) two BERT models fine-tuned to extract relations from Portuguese product titles in the smartphone/cellphone category; and (ii) a comparison between the extracted data showing how unstructured data can complement structured information.

This document is divided into five sections. Section 2 presents related work; Section 3 describes how the RE models were generated and evaluated, and discusses the results; Section 4 compares the extracted instances with a corpus built from structured data. Section 5 finishes this paper with some conclusions and proposals for future work.

2. Related Works

The Relation Extraction (RE) task consists in extracting well-defined relationships between two entities [Pawar et al. 2017] and saving them into a structured repository [Moens 2006, Sarawagi 2008]. Hearst [Hearst 1992] proposes lexical-syntactical patterns to identify relations. The ACE program [Doddington et al. 2004] aims to analyze other aspects in sentences, such as the occurrence of words and lexical categories. Over time, many works also considered named-entity recognizer models as a crucial part of the RE task [Sarawagi 2008] and vice-versa [Ji and Grishman 2006]. The task also became a subject of research in Machine Learning (ML) and NLP, where the main investigated approaches were Support Vector Machines [Zitouni and Florian 2008] and Conditional Random Fields [Li et al. 2011].

More recent studies showed promising results to RE using deep neural networks, such as Convolutional Neural Networks [Zeng et al. 2014] and Recursive Neural Networks [Socher et al. 2012, Hashimoto et al. 2013]. Deep contextualized language models, such as BERT [Devlin et al. 2019], have gained attention in ML and NLP tasks [Peters et al. 2018, Radford et al. 2018, Devlin et al. 2019], such as "Question Answering" [Devlin et al. 2019] and RE [Soares et al. 2019]. Thus, this work explores a fine-tuned BERT architecture for RE, as will be described in the next sections.

[2] `https://github.com/plkmo/BERT-Relation-Extraction`

2.1. Relation Extraction with BERT

The Bidirectional Encoder Representations From Transformers (BERT) [Devlin et al. 2019] is an encoder architecture for generating contextualized language models. The model is versatile, able to understand context on the left and right to solve various NLP tasks, such as Next Sentence Prediction, Question Answering and Sentiment Analysis [Devlin et al. 2019].

In [Soares et al. 2019] the authors used BERT to represent relations via training following the matching the blanks (MTB) approach. By applying BERT to the task of extracting binary relations between entities, the authors start from a corpus of blocks of text containing two marked entities as illustrated in Table 1.

Table 1. Examples of marked entities and its substitution to "blanks". Adapted from [Soares et al. 2019]

r_A	In 1976, e_1 (then of Bell Labs) published e_2 , the first of his books on programming inspired by the Unix operating system.
r_B	The "e_2" series spread the essence of "C/Unix thinking" with makeovers for Fortran and Pascal. e_1's Ratfor was eventually put in the public domain.
r_C	e_1 worked at Bell Labs alongside e_3 creators Ken Thompson and Dennis Ritchie.
Mentions	e_1 = Brian Kernighan, e_2 = Software Tools, e_3 = Unix

Henceforth, the training set is created by replacing the entity with a special symbol `[BLANK]` in order to predict the hidden entity. The symbol is introduced probabilistically to ensure that the model learns the relationship not only by the entities, but by the words around them. This process was called "matching the blanks". For the authors, MTB training aims to solve the data redundancy problem observed in texts on the web, where an arbitrary pair of entities is probably mentioned several times throughout a sequence.

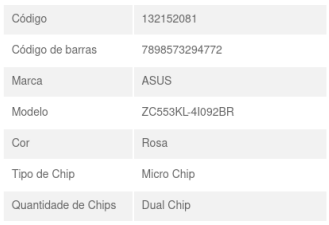
The authors propose a representation method called *entity markers*: given a sequence of tokens, starting with token `[CLS]` and ending with `[SEP]`, the tokens that mention a certain entity are delimited. For this, they used the $\text{BERT}_{\text{LARGE}}$ pre-trained model and Wikipedia in English as the training corpus, with interconnected paragraph blocks. In their experiments with the MTB method, the authors observed an F-score value of 89.5%, better than the 71.5% value that was observed for the TACRED [Zhang et al. 2017] relation prediction model on the SemEval 2010 dataset. In addition, the MTB obtained 89.2 10-way 1-shot[3] on the FewRel dataset, against 94.3% obtained from humans. Finally, it is worth mentioning that there is an open implementation of this work[4].

3. Experiments and Results

This section describes datasets, experiments and results. We used a dataset of products from the smartphone category (smartphones and cellphones)[5]. This dataset has instances

[3]This is a training method which contains 1 instance of a single class between 10 of them.
[4]`https://github.com/plkmo/BERT-Relation-Extraction/`
[5]This category was chosen because of its high demand on e-commerce platforms.

of structured information in product sheets (as shown in Figure 1) as well as unstructured information in product titles and descriptions (as shown in Figure 2)[6].



Código	132152081
Código de barras	7898573294772
Marca	ASUS
Modelo	ZC553KL-4I092BR
Cor	Rosa
Tipo de Chip	Micro Chip
Quantidade de Chips	Dual Chip

Figure 1. Example of a product's data sheet

Figure 2. Example of a product's title and description

This entire dataset contains 956 products from the smartphone category. It was separated in two sets: (i) one with 540 items with structured information (product sheets) and (ii) one with 416 product titles annotated with entities and binary relations.

Product sheets – From the 540 products, 77 different properties were recovered from their data sheets. Not all products have all properties. For example, the property called "*garantia do fornecedor*" (vendor guarantee) is present in all 540 products, while the property called "*conexões*" (connections) is only present in 201 products.

Annotated titles – 416 product titles were annotated using the Prodigy[7] tool by 2 linguists[8], who marked the following entities: `Model`, `Brand`, `Color`, `Internal_memory`, `Camera`, `Display_size`, `Chip_capacity`, `OS` (operating system) and `Processor`. Thus, each mention of a `Model` (subject) entity and an entity of another type (object) in the same title (that is, each pair of marked entities) becomes an instance of a binary relation of interest in the dataset. Examples of such relations include `has_brand(Model, Brand)` and `has_color(Model, Color)`. A total of 8 different relations were identified.

3.1. Experiments

Experiments were designed to answer the following research questions using the two datasets:

Q1 – How much relational information is it possible to extract from product sheets?
Q2 – How much relational information is it possible to extract from product titles?
Q3 – How complementary is the relational information extracted from titles to the one extracted from the product sheets?

To answer **Q1**, Subject-Predicate-Object (SPO) triples were constructed using properties extracted from the product sheets as well as their respective values. Therefore, the following design was adopted:

[6] `https://www.americanas.com.br/`. Last access: June 2021
[7] `https://prodi.gy/`
[8] Discrepancy cases were resolved by a third linguist, although the agreement rate between the annotators was above 72%.

- **Subject entity** – this is the value of a `Model` entity. If the product's sheet did not contain this attribute, a Named-Entity Recognizer (NER) trained in the e-commerce domain was used to recognize the `Model` entity from the product title. This NER was generated by another team linked to the partnership with americanas s.a.
- **Relation label** – this is one of the 8 relations of interest.
- **Object entity** – this is the value of the corresponding property in the product sheet. For example, `Full HD - 1920x1080` or `5.2"` may be values for the `has_display_size` relation. Similarly, `Android` is a possible value for the `has_os` relation.

In order to answer **Q2**, we trained the MTB [Soares et al. 2019] approach on product titles annotated with entities and relations. Following an implementation of MTB[9], each instance used in the model's fine-tuning consists of: (1) a sentence (in the case of this experiment, a product title) with two marked entities and (2) the label of the relation between them. The annotated titles dataset was split into training, validation and test partitions as detailed in Table 2.

Table 2. Relation instances on smartphone dataset and their distribution into training, validation and testing sets.

	train	valid	test	**total**
has_brand	199	103	103	405
has_camera	108	70	53	231
has_chip_capacity	124	63	66	253
has_color	170	89	92	351
has_display_size	117	67	68	252
has_internal_memory	127	77	73	277
has_os	68	39	40	147
has_processor	18	9	8	35
Total	931	517	503	1951

The original source code was adapted[10] to use models that are capable of dealing with Brazilian Portuguese:

- **BERTimbau**[11] [Souza et al. 2020] **–** this is a trained BERT model for Brazilian Portuguese based on web documents from various domains.
- **Multilingual BERT**[12] [Devlin et al. 2019] (mBERT) **–** this is a BERT model trained for more than 100 languages, including Portuguese, based on Wikipedia content[13].

These models were trained with batch size 128, MTB learning rate 10^4 and fine-tuning learning rate 7×10^5 (as suggested by the original implementation). Both models trained MTB within 18 epochs (approximately 3 days each model), while requiring 60 and 65 epochs (approximately 2 hours each model) to fine-tune BERTimbau and mBERT,

[9] `https://github.com/plkmo/BERT-Relation-Extraction`
[10] `https://github.com/joaobarbirato/BERT-Relation-Extraction`
[11] `https://huggingface.co/neuralmind/bert-base-portuguese-cased`
[12] `https://huggingface.co/bert-base-multilingual-uncased`
[13] More details on Multilingual BERT training are available at `https://github.com/google-research/bert/blob/master/multilingual.md`

respectively. All training steps were performed on a 40 core Intel(R) Xeon(R) Silver 4210 CPU 2.20GHz machine.

Finally, regarding **Q3**, a third experiment was carried out to compare the information extracted from structured (**Q1**) and unstructured (**Q2**) data. The same NER model used on **Q1** was used to process the 540 titles corresponding to each product used for **Q1** to automatically mark entities. These marked titles served as input to the MTB BERTimbau model for inferring the relations.

3.2. Results

To answer **Q1**, 2,825 model-attribute-value triples were extracted from the 540 product sheets. Table 3 shows some examples of relation instances extracted from product sheets. From the extracted relations it is possible to see that there is still room for improvement. For example, entities `Moto G (3ª Geração)` and `Moto G 3` were considered as different entities. Disambiguating entities is one possible solution to such problems.

Table 3. Examples of relation instances extracted from the product sheet dataset.

Relation	Subject	Object
has_internal_memory	SM-N975F/2DL	256gb
has_color	ZC554KL-4A115BR	preto
has_display_size	Galaxy S8	5.8”
has_camera	Moto G (3ª Geração)	13mp

To answer **Q2**, from the 503[14] instances in the test set, MTB models trained using BERTimbau and multilingual BERT (mBERT) correctly extracted, respectively: 378 and 376 instances. On average, the model trained using BERTimbau performed better regarding the F-score values, with 3.41 percentage points more than mBERT, as shown in Table 4. Indeed, in [Souza et al. 2020] the authors pointed out a similar difference between the F-score values for BERTimbau and mBERT.

Regarding **Q3**, the model from **Q2** was applied to the same dataset as **Q1** in order to compare the information extracted from structured and unstructured data. From the 540 items in the product sheet dataset, we processed the product titles to generate 4,933 inputs for the model trained with BERTimbau infer the relation instances. Since different titles can generate the same relation instance, from these titles, BERTimbau output 2,575 distinct triples. Comparing the extracted triples with the entities identified by the NER model we noticed that 2,072 were equal. We considered these as the correct ones although this decision may be ignoring the NER errors. Table 4 shows detailed results for each model, relation and research question.

The results regarding **Q2** indicate the applicability of BERT Relation Extraction to extract binary relations from product titles. The model trained using BERTimbau was selected to be used in our third experiment due to its very good F1-score (almost 94%).

One of the main reasons for the worse result in the experiment related to **Q3** compared to the one regarding **Q2** are the differences in quality and standardization between

[14]It is worth mentioning that different titles can generate the same relation instance. Of 503 product titles, BERTimbau and mBERT output 405 and 407 distinct relation instances, respectively.

Table 4. Evaluation values (%) (a) in test sets for the MTB models and (b) in Q1 dataset using the MTB BERTimbau trained model

	(a) **Q2**					(b) **Q3**		
		MTB BERTimbau		**MTB mBERT**			**MTB BERTimbau**	
Relation	**Support**	**Accuracy**	**F1**	**Accuracy**	**F1**	**Support**	**Accuracy**	**F1**
has_processor	8	87.50	**93.33**	62.50	66.67	476	50.84	66.30
has_os	40	90.00	92.31	90.00	**93.51**	605	77.85	79.63
has_internal_memory	73	100.00	97.99	100.00	**99.32**	15	80.00	4.57
has_display_size	68	89.71	94.57	92.65	**96.18**	759	74.18	83.90
has_color	92	98.91	**94.79**	97.83	91.37	645	94.26	84.04
has_chip_capacity	66	89.39	89.39	92.42	**93.85**	589	78.95	84.16
has_camera	53	100.00	**96.36**	100.00	95.50	1101	92.28	93.81
has_brand	103	90.29	**92.08**	85.44	87.13	743	75.24	81.84
Mean_{micro}	-	93.23	**93.85**	90.10	90.44	-	77.95	72.28

these two datasets. The titles used for **Q1** follow stricter standardization rules and quality requirements, as they refer to products sold by a single large e-commerce company. The titles used for the NER model training were provided by a diverse set of small sellers, and therefore are noisier and less standardized. We believe that this difference in data was responsible for the poor performance of the NER in this new dataset. We manually observed that the NER tagged many false instances of `Model`, which could have drastically affected many predicted relation instances.

4. Qualitative Analysis

In this section we compare the relation instances extracted from both datasets (structured and unstructured) to better understand how different and complementary are the triples extracted from them by comparing, respectively, results from **Q1** with **Q2** and **Q1** with **Q3**; thus answering **Q3**. Numbers verified in both analysis were obtained using set operations in code.

Table 5 quantifies the amount of instances extracted (**Q2** vs **Q1** – Different) and inferred (**Q3** vs **Q1** – Complementary). Columns (a) and (c) quantify the instances present only in **Q2** and **Q3**, respectively. The other columns quantify the instances that were present both in **Q2** and **Q1** (b) and **Q3** and **Q1** (d).

How different are they? From the 405 relation instances predicted by the BERTimbau model in **Q2**, 378 (approximately 93%) were correct. It was verified, then, how many of these correctly extracted instances were equal to the ones extracted from the product sheet dataset. Only 11 common instances were found. Consequently, about 97% of the correctly predicted instances (367 instances) are correct and new. In other words, it is possible to derive a lot of correct information from product titles that are not yet available in product sheets.

How complementary are they? Based on this information, it is possible to identify how the information in product titles complements the information found in product sheets. Only 202 (9.75%) of the $2,072$ correctly inferred triples in **Q3** were extracted from product sheets. Consequently, about 90.25% of the correctly predicted instances ($1,870$ instances) are correct and new. In other words, we again conclude that it is possible to derive a lot of correct information from product titles that are not yet available in product sheets.

Table 5. Amount of instances retrieved in the product sheets (Q1) in comparison with instances extracted by the BERTimbau model (Q2 and Q3)

	Q2 vs Q1 – Different		**Q3 vs Q1 – Complementary**	
Relation	**Only Q2** (a)	**Q2 ∩ Q1** (b)	**Only Q3** (c)	**Q3 ∩ Q1** (d)
has_color	74	2	406	73
has_brand	67	3	199	58
has_internal_memory	56	1	9	-
has_display_size	48	-	296	5
has_chip_capacity	46	-	206	1
has_camera	42	3	381	57
has_os	28	2	262	8
has_processor	6	-	111	-
Total	367	11	1,870	202

5. Conclusion

In this paper we investigated relation extraction from structured and unstructured data for the e-commerce domain using a BERT model fine-tuned for this task. We concluded that the fine-tuned model using BERTimbau performs a little better than the one based on Multilingual BERT. We compared how different and complementary are the information extracted from product titles and the structured information present in product sheets.

Experiments showed that about 97% of the relation instances extracted from an external dataset and 90.25% of the triples extracted from the same source were correct and new, i.e. not present in product sheets. From these experiments, we can conclude that processing unstructured data from product titles, which is much more abundant and easier to collect, is a promising approach for generating structured data that can be useful for a variety of e-commerce applications such as filtering and recommendation.

From the qualitative analysis, it is clear that the automatic relation extraction in a corpus of unstructured data composed of product titles contributes towards constructing a relation instance corpus. Evidently, the information on e-commerce is incomplete and the MTB method contributes to the completion of entity linkages.

As future work, it is possible to optimize MTB training hyperparameters, as this was not done due to implementation difficulties, integration with BERT models for Portuguese and training time. We also intend to use the extracted relation instances to build a knowledge graph (KG) and study its effectiveness in tasks for the e-commerce domain, such as product recommendation and search. The results presented in this paper support this idea, since most of the instances extracted by the MTB models were not in the base KG, which was built from structured data. This analysis shows that the relation extraction can help with the knowledge graph completion problem.

Acknowledgments

This paper and the research behind it would not have been possible without the support of americanas s.a. Digital Lab, specially José Pizani and Ester Campos, who closely followed this research. This work is is part of the project “Dos dados ao conhecimento: extração e representação de informação no domínio do e-commerce” (Projeto de extensão - UFSCar #23112.000186/2020-97).

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. M., and Weischedel, R. M. (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Hashimoto, K., Miwa, M., Tsuruoka, Y., and Chikayama, T. (2013). Simple customization of recursive neural networks for semantic relation classification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1372–1376.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Ji, H. and Grishman, R. (2006). Analysis and repair of name tagger errors. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 420–427.

Li, Y., Jiang, J., Chieu, H. L., and Chai, K. M. A. (2011). Extracting relation descriptors with conditional random fields. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 392–400.

Moens, M.-F. (2006). *Information extraction: algorithms and prospects in a retrieval context*, volume 21. Springer Science & Business Media.

Pawar, S., Palshikar, G. K., and Bhattacharyya, P. (2017). Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding with unsupervised learning.

Sarawagi, S. (2008). Information extraction. *Found. Trends Databases*, 1(3):261–377.

Soares, L. B., FitzGerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Socher, R., Huval, B., Manning, C. D., and Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., and Achan, K. (2020). Product knowledge graph embedding for e-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 672–680, New York, NY, USA. Association for Computing Machinery.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 2335–2344.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. (2017). Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

Zitouni, I. and Florian, R. (2008). Mention detection crossing the language barrier. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 600–609.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Classificação multimodal para detecção de produtos proibidos em uma plataforma *marketplace*

Alan da Silva Romualdo[1], **Livy Real**[2], **Helena de Medeiros Caseli**[1]

[1]Universidade Federal de São Carlos (UFSCar) – Departamento de Computação
Caixa Postal 676 – 13565-905 – São Carlos – SP – Brasil

[2]Digital Lab – americanas s.a.
São Paulo – SP – Brasil

alan.romualdo@b2wdigital.com, helenacaseli@ufscar.br

livy.coelho@b2wdigital.com

Abstract. *The multimodal learning aims to explore the characteristics of different modalities (text, image, audio) to generate computational models. In electronic commerce, due to the great variety of product features and the absence or inconsistency of information, the combination of information from different modes is quite adequate. This work presents some experiments caried out for the multimodal classification (text and image) of products (adult products) that cannot be sold in the marketplace of the partner company. In these experiments, neural networks were used to train uni and multimodal classifiers. The multimodal classifier achieved* 99% *of* $F1$ *against* 98% *for the textual model and* 94% *for the visual one.*

Resumo. *O aprendizado multimodal visa explorar as características das diversas modalidades (texto, imagem, áudio) para gerar modelos computacionais. No comércio eletrônico, devido à grande variedade das características dos produtos e à ausência ou inconsistência de informações, a combinação de informações de modos diferentes vem a ser bastante adequada. Neste trabalho são apresentados alguns experimentos para a classificação multimodal (texto e imagem) de produtos (produtos adultos) que não podem ser vendidos no marketplace da empresa parceira. Nesses experimentos, redes neurais foram usadas para treinar classificadores uni e multimodal. O classificador multimodal atingiu* 99% *de* $F1$ *contra* 98% *do modelo textual e* 94% *do visual.*

1. Introdução

O comércio eletrônico no Brasil acaba de bater recorde com um faturamento de R$$53,4$ bilhões no primeiro semestre de 2021, segundo dados da 44^{a} edição do Webshoppers, relatório elaborado pela Ebit | Nielsen[1]. Nos últimos anos, a sua importância para as empresas e a população em geral cresceu consideravelmente. Uma das razões para esse crescimento são os *marketplaces*, que durante o isolamento na situação da pandemia (entre 2020-2021) tornaram-se uma alternativa fundamental para que muitas lojas, inclusive supermercados, continuassem faturando. Segundo a Associação Brasileira de Comércio

[1]https://www.ecommercebrasil.com.br/noticias/e-commerce-no-brasil-bate-recorde-e-atinge-r-53-bilhoes-ebit-nielsen-webshoppers/

Eletrônico (ABCOMM)[2], em 2020, os *marketplaces* foram responsáveis por 78% do faturamento total do comércio eletrônico.

Em um *marketplace*, o modelo de negócio permite que um vendedor insira na plataforma de venda as informações, imagens ou descrições dos produtos personalizadas para seus anúncios. Junto com essa maior flexibilização surgem também alguns desafios como ter que lidar com a falta de informações, imagens que não satisfazem o padrão do *marketplace* e textos descritivos com especificações não estruturadas. Para tentar contornar esses problemas, algumas empresas utilizam de corretores automáticos, verificação/curadoria manual, categorizações manuais ou automáticas, ferramentas para classificação, etc.

Para realizar essa verificação/curadoria, os *marketplaces* geralmente contratam empresas especializadas ou usam plataformas de *crowdsourcing* para classificar manualmente os produtos. Contudo, devido à grande quantidade de novos produtos carregados diariamente e à natureza dinâmica das categorias, as soluções de aprendizado de máquina surgem como uma alternativa para classificar automaticamente os produtos e reduzir o custo desta tarefa [Zahavy et al. 2018].

Nesse cenário, a classificação correta de um produto é fundamental não apenas para garantir a visibilidade do produto, mas também para determinar se ele satisfaz as políticas de venda do *marketplace*. Este trabalho foca especificamente nesse último ponto, apresentando soluções para classificar automaticamente os produtos cadastrados pelos vendedores com o intuito de barrar aqueles que não podem ser vendidos devido às políticas do *marketplace* (por exemplo, produtos ilegais).

Como as informações dos produtos estão, em sua maioria, na forma de dados não estruturados nas modalidades de texto e imagem, este trabalho investigou como modelos uni e multimodal se saem na classificação de produtos da categoria Adulto.[3] A hipótese investigada neste trabalho é a de que os métodos que lidam com o aprendizado em mais de uma modalidade têm o potencial de enriquecer a representação dos produtos, tornando possível uma melhora no desempenho da tarefa de classificação em relação aos modelos unimodais. Para tanto, foram utilizados os conjuntos de dados de produtos contendo informações textuais (títulos e descrições) e visuais (fotos e imagens) fornecidos pela empresa parceira deste projeto.

As principais contribuições deste trabalho são: (i) análise de desempenho da classificação multimodal de produtos que não devem ser vendidos no *marketplace* da empresa parceira, e (ii) geração de um modelo multimodal com alto desempenho e pronto para entrar em produção.

O restante deste artigo está organizado como segue. Na Seção 2 são apresentados alguns trabalhos da literatura selecionados como os mais relevantes para tratar o problema de classificação multimodal para o *e-commerce*. A Seção 3 descreve o conjunto de dados (3.1) e os modelos unimodal textual (3.2), visual (3.3) e multimodal (3.4) desenvolvidos neste trabalho. A avaliação desses modelos é apresentada na Seção 4 e a Seção 5 encerra este documento com algumas conclusões e propostas de trabalhos futuros.

[2]`https://abcomm.org/noticias/marketplaces-crescimento-exponencial-ao-longo-da-pandemia/`

[3]A categoria Adulto contém produtos adequados para maiores de 18 anos como itens relacionados a sexo.

2. Trabalhos relacionados

O aprendizado de máquina no contexto do comércio eletrônico, assim como em diversos outros cenários reais, enfrenta desafios para lidar com uma distribuição irregular de dados. Uma das estratégias adotadas para tentar solucionar esse problema é combinar as informações vindas de diferentes modalidades, como o textual e o visual, no que chamamos de aprendizado multimodal [Bi et al. 2020]. Segundo [Peng et al. 2018], as características específicas de cada modalidade levam a uma heterogeneidade na representação, o que faz com que seja necessário refinar as modalidades para que os métodos de aprendizado multimodal não aprendam características erradas ou prejudiciais para o modelo.[4] Além disso, há a dificuldade relacionada a como aprender essas representações de maneira conjunta.

No aprendizado multimodal, as informações em um modo (por exemplo, o textual) são combinadas com as informações em outro modo (por exemplo, o visual) via um processo de fusão. Os tipos de fusão podem ser divididos com base no momento em que a fusão ocorre, podendo haver fusão no início e no fim do processamento das modalidades, e são agrupados em: fusão em nível de recurso (*early fusion*) ou fusão em nível de decisão (*late fusion*) [Zahavy et al. 2018].

Em [Bi et al. 2020], os autores utilizaram as duas estratégias e a fusão em nível de decisão obteve melhor desempenho ($F1 = 90,94\%$). Em [Chordia and Kumar 2020], os autores também fazem uso de modelos para cada modalidade específica, mas utilizam também um técnica de *co-attention* proposta em [Lu et al. 2016] , à qual atribuem uma importante contribuição para a performance geral de sua proposta ($F1 = 91,36\%$).

Os autores de [Wirojwatanakul and Wangperawong 2019] também utilizaram a abordagem *late fusion* para categorizar produtos à venda na Amazon. Eles treinaram modelos para as modalidades específicas separadamente (texto, imagem, descrição) que chamaram de modelo de fusão tri-modal. Embora tenham obtido bons resultados ($F1 = 88,2\%$) os autores apontaram um número significante de erros e um direcionamento para extensão dos dados em trabalhos futuros.

Em [Zahavy et al. 2018], os autores fizeram uma arquitetura com 3 componentes: (1) uma CNN de [Kim 2014] para o texto, (2) uma CNN de [Simonyan and Zisserman 2015] para imagem e (3) uma rede neural de decisão, que aprende a escolher qual classificação considerar entre essas duas modalidades.

O que se pode resumir da breve análise dos trabalhos relacionados apresentada nesta seção, é que, no geral, todos os modelos foram desenvolvidos para modalidades de texto e imagens, utilizando de CNNs para imagens, como VGG [Simonyan and Zisserman 2015], ResNet [He et al. 2016] ou CNN de [Kim 2014]; e alguns métodos de PLN, como BERT [Devlin et al. 2018] ou *embeddings* gerados por métodos bastante conhecidos na área, como Glove [Pennington et al. 2014], para textos.

3. Experimentos

Esta seção descreve o conjunto de dados e os modelos uni e multimodal utilizados nos experimentos.

[4]Exemplos de tarefas para o refinamento nas modalidades são: remoção de *stop-words*, números ou caracteres especiais nos textos; e ajuste de regiões de interesse ou aplicação de filtros nas imagens.

3.1. Conjunto de dados

Os dados utilizados nos experimentos apresentados neste artigo são referentes a 8.668 produtos, sendo 4.334 de conteúdo categorizado como Adulto (classe positiva, 1) e outros 4.334 produtos permitidos à venda (classe negativa, 0).

Cada instância possui: (i) uma imagem que representa o produto, (ii) seu título e (iii) sua descrição. A Figura 1 traz dois exemplos desse conjunto de dados, um da classe positiva (à esquerda) e outro da classe negativa (à direita). Esse conjunto de dados foi dividido em 60% para treinamento, 30% para validação e 10% para teste. Resultaram, então, 5.202 produtos para treinamento, 2.602 para validação e 864 de teste, todos igualmente distribuídos para as duas classes.

Título: A Peça Do Adulto Do Jogo Do Sexo De 4 Partes Ajuda A Dados Da Posição Do Sexo Para Preliminares Do Quarto Dos Pares

Descrição: Jogo de Dados Sexuais Dados Eróticos de Amor Adicionam Especiarias para Casais Solteiros Adultos. Faça um aniversário de aventureiro ou [...]

Título: Cinta Colete Corretor Postura Coluna Lombar tamanho XXL=EXTRA G

Descrição: Cinta Colete Corretor Postura Coluna Lombar tamanho XXL=EXTRA G

Figura 1. Exemplos de produtos da classe positiva e negativa

3.2. Modelo textual

Para a modelagem textual (títulos e descrições) foram geradas *word embeddings* usando o FastText [Bojanowski et al. 2017], abordagem CBOW, com 64 dimensões, sendo considerada a média dos vetores das palavras para a representação da sentença.[5] Para a geração desses *embeddings* foi utilizado um *corpus* composto por títulos e descrições de aproximadamente 7,5 milhões de produtos.

Antes de obter as *word embeddings* pelo FastText (3), todos os textos (1) são pré-processados a fim de remover *stopwords*, caracteres especiais, sequências numéricas, links e realizar a conversão para minúsculo (2), como ilustrado no exemplo abaixo.

```
1. Faca Esportiva Xingu XV2562 Outdoor com Bainha eBússola Camuflada
2. faca esportiva xingu xv outdoor bainha bussola camuflada
3. [ 0.01715468, -0.01149213, 0.03243327, ... ,-0.06800657, -0.03518752]
```

Nos experimentos para a classificação de itens da categoria Adulto foram gerados dois modelos textuais: um apenas para títulos e outro para títulos e descrições. No modelo textual de título, a entrada é o vetor de 64 dimensões das *embeddings* do título. Já no

[5]O FastText foi escolhido por ter sido o de melhor desempenho em experimentos preliminares para cálculo de similaridade multimodal com dados de comércio eletrônico, em comparação com Glove [Pennington et al. 2014] e Word2Vec [Mikolov et al. 2013]. Vale mencionar que as *word embeddings* geradas usando o FastText também foram melhores quando comparadas com as de domínio geral do NILC [Hartmann et al. 2017].

modelo de título e descrições, a entrada são dois vetores concatenados onde primeiro é do título e o segundo, das descrições, totalizando um vetor de tamanho 128 como ilustrado na Figura 2. Por não haver *overfitting* no treino dos modelos textuais, optou-se por não adicionar uma camada de regularização (dropout).

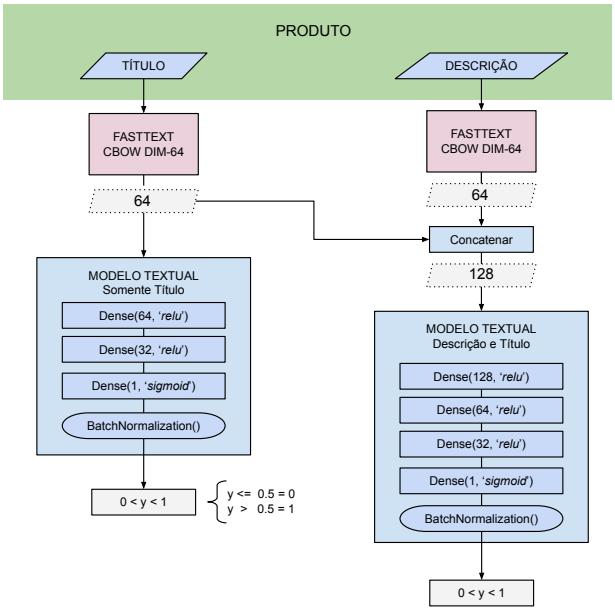


Figura 2. Descrição dos modelos unimodais textuais, onde a saída (y) representa a probabilidade do produto ser Adulto.

Para usar o modelo para a classificação pode-se considerar, por exemplo, que dado um y maior do que $0,5$ indica um produto que deve ser considerado Adulto (saída/classe igual a 1) e que abaixo de $0,5$ seja não Adulto (saída/classe igual a 0).

Todos os modelos unimodais foram implementados utilizando o Keras (v2.5.0) [Chollet et al. 2015] e seus hiper-parâmetros (como *batch-size*, *loss* e outros) foram escolhidos de maneira empírica O treinamento foi realizado com 350 épocas, tamanho do *batch* igual a 32 e uma taxa de aprendizado Adam de 1×10^{-5}, *loss binary cross-entropy*, com um total 6.277 parâmetros em cada modelo. A máquina utilizada para o treinamento dos modelos possui Windows 10 build 20H2, processador AMD Ryzen 5 3600 6-Core Processor com 12 núcleos de até 3.6GHz, 16GB de memória RAM e placa de vídeo (GPU) NVIDIA GeForce RTX2060 de 6GB de VRAM dedicado. O tempo de treinamento de cada modelo ficou em torno de 6 minutos.

3.3. Modelo visual

No modelo visual, ilustrado na Figura 3, todas as imagens foram redimensionadas para 224×224 pixeis, mantendo as 3 dimensões que representam as cores RGB. Alguns produtos possuíam mais de uma imagem, mas nos experimentos descritos neste artigo optou-se por limitar apenas à uma imagem.

Para esse modelo visual também utilizou-se o Keras [Chollet et al. 2015], que possui diversos modelos de redes neurais convolucionais pré-treinados em *datasets* como ImageNet [Deng et al. 2009], MNIST [LeCun and Cortes 2010] e CIFAR [Krizhevsky 2012]. Para a construção desse modelo, técnicas como *transfer learning* e *fine tuning* foram utilizadas para otimizar a extração de características e o aprendizado.

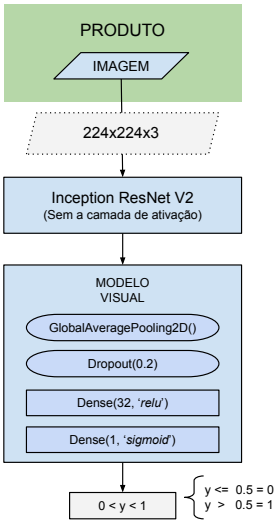


Figura 3. Descrição da combinação do modelo pré-treinado Inception ResNet V2 concatenado ao nosso modelo visual com função de ativação *sigmoid*.

Para o *transfer learning*, utilizou-se a rede Inception ResNet V2 [Szegedy et al. 2017] pré-treinada para o ImageNet de 1.000 classes. Sua arquitetura possui 780 camadas e a camada de classificação foi removida e concatenada ao modelo visual desse experimento. Para o *fine tuning*, foi feito um congelamento da camada inicial até a camada 650, a fim de garantir que o modelo faça a mesma extração de característica do seu pré-treinamento. O modelo visual (Figura 3) gerado dessa maneira é, então, aplicado para fazer a classificação com base nas *features* de uma nova imagem extraídas pela rede Inception ResNet V2.

O treinamento foi realizado com 25 épocas, tamanho do *batch* igual a 16 e uma taxa de aprendizado Adam de 1×10^{-5}, e *loss binary cross-entropy*, com um total $55.873.736$ parâmetros. A máquina usada para o treinamento desse modelo visual foi a mesma usada no modelo textual e o treinamento levou cerca de 18 minutos.

3.4. Modelo multimodal

O modelo multimodal utiliza os modelos unimodais pré-treinados sem as suas camadas de ativação. Os modelos são concatenados e, então, transportados por camadas densas. A função de ativação também é a *sigmoid* e para o treinamento foram configuradas 100 épocas, tamanho de *batch* igual a 32, *loss binary cross-entropy* e uma taxa de aprendizado Adam de 1×10^{-4}. O módulo visual é congelado até a camada 650 e o textual é congelado até a camada `Dense(32, 'relu')` (veja Figura 2).[6]

[6]Novamente, a mesma máquina foi usada para o treinamento do modelo multimodal por cerca de 1 hora e meia.

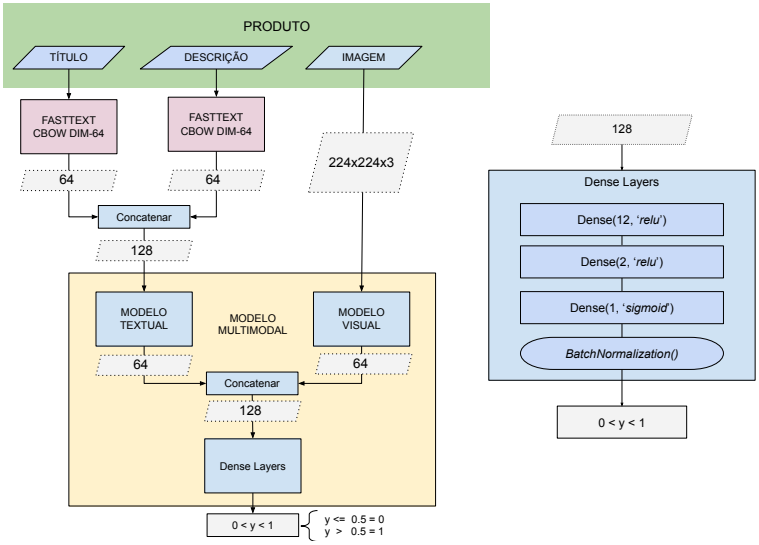


Figura 4. Descrição do modelo multimodal textual-visual produzido para este experimento.

4. Resultados

Os valores para as medidas de avaliação precisão, cobertura e $F1$, para os modelos unimodais e multimodal, para cada classe, são apresentados na Tabela 1.

Tabela 1. Valores das medidas de avaliação para os modelos unimodais de texto e de imagem e multimodal

modelo	classe	precisão	cobertura	$F1$
`textual-título`	0	93%	97%	95%
`textual-título`	1	96%	93%	94%
`textual-título+descrição`	0	98%	98%	98%
`textual-título+descrição`	1	98%	98%	98%
`visual`	0	94%	94%	94%
`visual`	1	94%	94%	94%
`multimodal`	0	99%	99%	99%
`multimodal`	1	99%	99%	99%

O modelo visual foi o que apresentou pior desempenho quando comparado aos demais. Acredita-se que os modelos textuais podem dar foco em palavras específicas e, com isso, podem ocasionar a classificação correta de um produto adulto. Já no visual, a variação de imagens de diferentes tipos de produtos pode ter trazido ruído para o aprendizado. Contudo, vale ressaltar que os modelos unimodais, quando fundidos no modelo multimodal, levaram a um aumento nos valores das medidas, indicando que os mesmos são complementares e comprovando nossa hipótese.

Para facilitar a análise qualitativa dos resultados, uma interface de visualização foi implementada. Nessa interface, ilustrada na Figura 5, são exibidos os erros associados aos modelos que erraram (indicado pelas cores). Por exemplo, o primeiro produto foi classificado de modo errado como positivo (produto adulto) por todos os modelos, enquanto o

terceiro exemplo também foi equivocadamente classificado como positivo pelos modelos visual e multimodal. Cada exemplo é acompanhado da classe correta (entre colchetes), da classe atribuída de modo errado pelo classificador, e as informações do produto usadas na classificação: título, descrição e imagem.

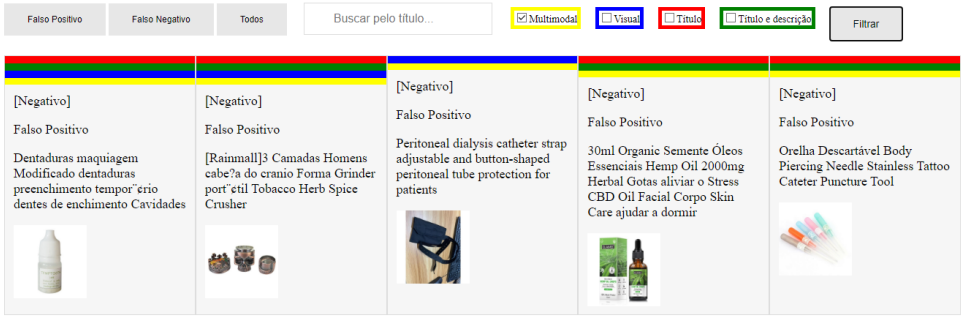


Figura 5. Interface desenvolvida para análise qualitativa dos erros de classificação dos modelos treinados ilustrando alguns falsos positivos.

Apenas 4 produtos que não podem ser vendidos no *marketplace* da empresa parceira foram classificados de forma errada (falso negativo) por todos os modelos.[7] Após analisar os possíveis motivos desse erro, notou-se que algumas informações textuais desses produtos estavam em inglês. Essa diferença de idioma pode ter sido prejudicial porque as *word embeddings* não foram treinadas para produtos nessa língua. Também foi possível observar que geralmente os produtos adultos possuem palavras específicas que podem ser utilizadas para classificar o produto nessa categoria. Dessa observação surge a proposta de retreinar o modelo do FastText incluindo produtos proibidos para venda, pois as *word embeddings* foram treinadas apenas a partir de produtos que estão disponíveis para a venda.

5. Conclusões e Trabalhos futuros

Este trabalho avaliou a classificação multimodal de produtos que não devem ser vendidos no *marketplace* da empresa parceira. Nos experimentos, constatou-se que o modelo unimodal de título e descrição apresentou um resultado muito bom ($F1 = 98\%$) mas sua combinação com o modelo visual, no modelo multimodal, foi ainda melhor ($F1 = 99\%$).

Como propostas de trabalhos futuros, tem-se: (i) retreinar o modelo do FastText incluindo itens do conjunto de produtos proibidos para venda; (ii) investigar a abordagem *ensemble* dos modelos e outras opções de fusão; (iii) estender os experimentos para outras categorias de produtos proibidos e (iv) colocar o modelo gerado em produção no *marketplace* da empresa parceira.

Agradecimentos

Esse artigo e a pesquisa desenvolvida não seriam possíveis sem o apoio da americanas s.a. Digital Lab, especialmente o apoio de José Pizani, Ester Campos e Jonas Ferreira. Esse trabalho é parte do projeto “Dos dados ao conhecimento: extração e representação de informação no domínio do e-commerce” (Projeto de extensão - UFSCar #23112.000186/2020-97).

[7]Essas instâncias não são apresentadas neste artigo por considerarmos seu conteúdo impróprio para o público em geral.

Referências

Bi, Y., Wang, S., and Fan, Z. (2020). A multimodal late fusion model for e-commerce product classification. *Proceedings of The 2020 SIGIR Workshop On eCommerce*, abs/2008.06179.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Chordia, V. and Kumar, V. (2020). Large scale multimodal classification using an ensemble of transformer models and co-attention. *CoRR*, abs/2011.11735.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hartmann, N. S., Fonseca, E. R., Shulby, C. D., Treviso, M. V., Rodrigues, J. S., and Aluísio, S. M. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In *Anais do XI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 122–131, Porto Alegre, RS, Brasil. SBC.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Krizhevsky, A. (2012). Learning multiple layers of features from tiny images. *University of Toronto*.

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

Lu, J., Yang, J., Batra, D., and Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 289–297, Red Hook, NY, USA. Curran Associates Inc.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Peng, Y., Qi, J., and Yuan, Y. (2018). Modality-specific cross-modal similarity measurement with recurrent attention network. *Trans. Img. Proc.*, 27(11):5585–5599.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4278–4284. AAAI Press.

Wirojwatanakul, P. and Wangperawong, A. (2019). Multi-label product categorization using multi-modal fusion models. *CoRR*, abs/1907.00420.

Zahavy, T., Krishnan, A., Magnani, A., and Mannor, S. (2018). Is a picture worth a thousand words? a deep multi-modal architecture for product classification in e-commerce. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Measuring Brazilian Portuguese Product Titles Similarity using Embeddings

Alan da Silva Romualdo[1], **Livy Real**[2], **Helena de Medeiros Caseli**[1]

[1]Federal University of São Carlos – Computing Department
Caixa Postal 676 – 13565-905 – São Carlos – SP – Brazil

[2]Digital Lab – americanas s.a.
São Paulo – SP – Brazil

alan.romualdo@b2wdigital.com, helenacaseli@ufscar.br

livy.coelho@b2wdigital.com

Abstract. *Textual similarity deals with determining how similar two pieces of texts are, considering the lexical (surface forms) or semantic (meaning) closeness. In this paper we applied word embeddings for measuring e-commerce product title similarity in Brazilian Portuguese. We generated some domain-specific word embeddings (using Word2Vec, FastText and GloVe) and compared them with general-domain models (word embeddings and BERT models). We concluded that the cosine similarity calculated using the domain-specific word embeddings was a good approach to distinguish between similar and non-similar products, but the multilingual BERT pre-trained model proved to be the best one.*

1. Introduction

The importance of e-commerce for companies and the general population has grown in recent years and even more in 2020. According to the Brazilian Association of Electronic Commerce (ABComm)[1], there was a growth of 56.8% in the first half of 2020 compared with the first eight months of 2019, with a turnover of approximately 8 billion dollars. With the global situation of the pandemic due to the SARS-CoV-2 virus, there was a major migration from physical stores to digital media. According to Mastercard's Global Outlook 2021 report[2], it is expected that 20-30% of the operations that migrated to digital media during social isolation become permanent.

According to [Rodrigues et al. 2014], the fast growth of the internet and e-commerce had made many companies to see them as a very interesting way to expand their business. In addition to the several marketing advantages, such as the dynamic trading and the reduction of marketing costs, in the online environment there is a direct large-scale exposure of products for sale. These characteristics favor communication and assortment global dissemination and contribute to the evolution of logistics, tending to reach a broader population.

[1]https://www.ecommercebrasil.com.br/noticias/faturamento-do-e-commerce-brasileiro-2020/

[2]https://www1.folha.uol.com.br/mercado/2021/01/ate-30-do-aumento-do-comercio-eletronico-relacionado-a-covid-deve-ser-permanente.shtml

In e-commerce, advertisements are usually composed of images and texts used to illustrate and to describe the products for sale. In marketplaces[3], typically the products information come from the vendors. In this case, the information is normally presented in a non-standard format and with high variability in terms of specifications and characteristics described for similar products, making it difficult to group them even for human beings.

For the automatic matching of similar products, it is necessary to use text and/or image processing techniques that are capable of extracting relevant characteristics for measuring the similarity between products. In this paper, we only deal with the **textual similarity** and, therefore, we apply natural language processing (NLP) techniques capable of finding similar products based on their textual information.

To illustrate this problem, consider the products shown in Figure 1[4]. In this figure there are: a pair of similar products (the first two) which are both ink cartridge of the same color (magenta) for the same printer, an in-class product (a kit of cartridges) and an out-class product (a printer).

Figure 1. Example of products that should be considered similar (the first two), another one from the same product category (the third one) and a non-similar product (the fourth one).

This paper examines the hypothesis that the textual similarity calculated based on the semantic distance between product titles can be applied for finding similar products in a marketplace such as Americanas[5]. By proving this hypothesis, similar products could be matched together before being offered as options in response to a customer query, thus improving shopping experience. In this paper we investigate product titles similarity based on **word embeddings** and BERT pre-trained models.

The main contributions of this work are: (i) the evaluation of the applicability of different word embedding and contextualized language models in measuring textual similarity in the e-commerce domain; and (ii) the addressing of a poorly explored scenario of e-commerce for Brazilian Portuguese.

This work is organized as follows. Section 2 presents some Related Works for calculating textual similarity in specific and general domains. Section 3 describes the investigated approaches, the *corpus* used in our experiments and the experimental setup.

[3]Marketplaces are online platforms that gather sellers offering different products or services.
[4]Image taken from `https://www.americanas.com.br/` accessed in 06/15/2021.
[5]`http://www.americanas.com.br`

Section 4 presents the results which pointed out that cosine similarity calculated using multilingual pre-trained BERT model achieved the best discrepancy ability. Finally, section 5 closes this paper with some conclusions and proposals for future work.

2. Related Works

In [Alam et al. 2020], the authors present several relevant approaches for calculating textual similarity in the field of biomedicine, including cosine similarity using word embeddings generated by GloVe [Pennington et al. 2014], Word2Vec [Mikolov et al. 2013] and FastText [Bojanowski et al. 2017]. They concluded that the general-domain word embeddings built by those tools did not work well at the sentence or paragraph level in the field of biomedicine because they did not capture medical terms neither optimized the word embeddings for the specific domain. According to these authors, similarity measuring techniques for a specific domain must take into account the semantic relevance of the information in that domain since misinterpretations about the content can lead the experts to bad decisions.

In [Lo 2017], word embeddings were also used for calculating the lexical and structural similarity for all language pairs. By means of Word2Vec [Mikolov et al. 2013] and other topic analysis tools, the authors concluded that their new version of MEANT was a more accurate alternative to BLEU [Papineni et al. 2002] in evaluating translation quality for low-resource languages.

In [Rosa da Silva et al. 2017], the problem of categorizing offers in the context of price comparison sites was investigated. They compared two techniques for generating word embeddings: one that learns unsupervised word embeddings from millions of offer descriptions (using BOW), and another that learns supervised word inclusion using a convolutional neural network (CNN). The CNN model substantially outperformed their best BOW model.

According to [Aryal et al. 2019] and [Zhang et al. 2020], there are several effective ways to calculate textual similarity using word embeddings, but the most traditionally used measures are the **cosine similarity** and the Euclidean distance. These measures calculate the degree of similarity between two objects based on the coordinates of these objects in a vector space [Alam et al. 2020, Arts et al. 2017].

Recently, a new measure for automatic evaluation in text generation was proposed: the BERTScore [Zhang et al. 2020]. Similar to other measures, BERTScore calculates a similarity score for each token in the candidate sentence with each token in the reference sentence using previously trained contextualized representations from a BERT model. According to [Zhang et al. 2020], BERTScore showed a better correlation with human judgments and a better model selection performance than other measures used in comparison.[6] Also according to these authors, BERTScore proved to be more robust in challenging examples compared to other evaluated measures.

In this paper, we present experiments carried out to evaluate how word embeddings and contextualized language models perform in measuring the similarity between

[6]The evaluation was made by comparing BERTScore with the following measures: BLEU, METEOR, ROUGE-L, CIDER, SPICE, LEIC, BEER, EED, CHRF ++ and CHARACTER. See [Zhang et al. 2020] for details.

product titles in Brazilian e-commerce.

3. Experiments

In this work, we investigated the most applied approaches for textual similarity measurement: word embeddings and contextualized language models. For that, different word embeddings models for the specific domain of e-commerce were generated using Word2Vec [Mikolov et al. 2013], FastText [Bojanowski et al. 2017] and GloVe [Pennington et al. 2014]. Pre-trained general domain word embeddings and BERT [Devlin et al. 2019] models available for Portuguese were also used to compare the results.

3.1. Experimental setup

For training the domain-specific (e-commerce) word embedding models, a *corpus* granted by Americanas was used, containing about 7.490 million products, with titles and descriptions totaling approximately 8 billion words. A vocabulary of $455,031$ words was extracted from this *corpus* containing the words that occur at least 2 times in the whole *corpus*.

Using this *corpus*, we trained five **domain-specific WEs** using 30 training epochs, a learning rate of 0.025 and word embeddings dimension equal to 64[7]:

1. `FastText-spec SKIPGRAM` – FastText word embeddings trained using SkipGram, Americanas *corpus* and character ngram maximum size of 6;
2. `Word2Vec-spec SKIPGRAM` – Word2Vec word embeddings trained using SkipGram and Americanas *corpus*;
3. `FastText-spec CBOW` – FastText word embeddings trained using CBOW, Americanas *corpus* and character ngram maximum size of 6;
4. `Word2Vec-spec CBOW` – Word2Vec word embeddings trained using CBOW and Americanas *corpus*;
5. `Glove-spec` – GloVe word embeddings trained with Americanas *corpus*.

In addition to these five domain-specific word embeddings, six other **general-domain** were used in comparison, all of them with dimension equal to 300 and trained by NILC[8] [Hartmann et al. 2017]:

7. `FastText-NILC SKIPGRAM` – FastText word embeddings trained using SkipGram;
8. `Word2Vec-NILC SKIPGRAM` – Word2Vec word embeddings trained using SkipGram;
9. `FastText-NILC CBOW` – FastText word embeddings trained using CBOW;
10. `Word2Vec-NILC CBOW` – Word2Vec word embeddings trained using CBOW;
11. `Glove-NILC` – GloVe word embeddings;

Finally, we also used **BERT models** for Portuguese: the multilingual BERT[9] and the BERTimbau [Souza et al. 2020] Large and Base[10] models :

[7]It is worth mentioning that we also trained word embeddings with dimension equal to 300 but the results were worse.

[8]Available at: `http://www.nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc`

[9]Available at: `https://github.com/google-research/bert`

[10]Available at: `https://github.com/neuralmind-ai/portuguese-bert`

12. `mBERT` – multilingual BERT model trained for 104 languages, including Portuguese.
13. `BERTimbau Base` – BERT model trained for Portuguese, with 12 layers and $110M$ of parameters.
14. `BERTimbau Large` – BERT model trained for Portuguese, with 24 layers and $335M$ of parameters.

It is worth mentioning that it was not possible to train a BERT model for our e-commerce big *corpus* with the available hardware. On the machine available for the experiments, which has 126GB of RAM and 16 processing cores, but no GPU, the estimated training time was more than 120 days.

3.2. Test *corpus*

As previously mentioned, the experiments presented in this paper aim to find titles of similar products by means of static or dynamic, domain-specific or general-domain embeddings. To assess this task, we chose to work with different levels of similarity by dividing our test *corpus* into four sets each one with 100 pairs of product titles:

- `manual` – this set contains 100 pairs of product titles that have been manually marked as similar;
- `automatic` – this set contains 100 pairs of product titles that were marked as similar by a simple automatic pattern matching system;
- `in-class` – this set contains 100 pairs of product titles that are not similar, but belong to the same product category;
- `out-class` – this set contains 100 pairs of product titles selected at random and manually checked to ensure they were not similar and were not even in the same category.

In Table 1 we present some examples of pairs of product titles in each of these classes.

Table 1. Examples of product titles for each of the test classes.

Product title	Class
Cartucho de tinta epson t196320 magenta xp204/xp401 -t196320 Cartucho de tinta epson T196320 magenta P/XP104/XP204/XP401	`manual`
Cartucho Epson 196 magenta T196 320BR 5 ml Cartucho Epson 196 Preto 5ml T196120	`automatic`
Kit Refil Tinta Com 04 Cores Epson L3110 L3150 T544 Epson Original 544 K M Y C Cartucho de Tinta HP 664 Preto - F6V29AB	`in-class`
Cartucho de Tinta HP 662 Preto - CZ103AB - HP Impressora Multifuncional HP Ink Advantage 2776 Jato de Tinta Wi-Fi - Impressora + Copiadora + Scanner	`out-class`

The product titles in the `manual` class are both for Epson cartridge (*cartucho*), with the same model (t196320) and color (*magenta*). The product titles in the `automatic` class are also of a Epson cartridge 196 but for different colors (magenta and black, *preto*). The `in-class` products are, respectively: a ink refill kit (*kit refil tinta*) and a cartridge. Finally, the `out-class` products are, respectively: a cartridge and a multifunctional printer (*impressora*).[11]

[11]The dataset was built by Americanas and it is a proprietary dataset.

4. Results

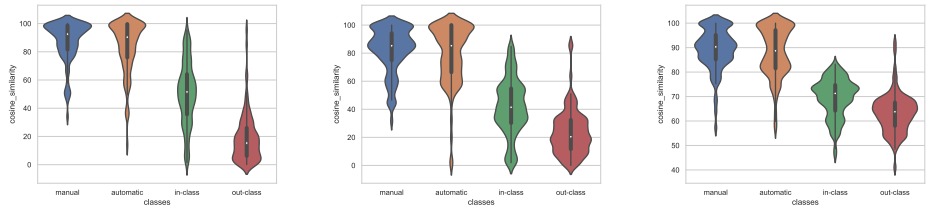
First, the average values of cosine similarity calculated using domain-specific word embeddings were compared with each other. Table 2 sumarizes the average cosine similarity values calculated using each domain-specific word embedding for each test set.

Table 2. Average values of cosine similarity calculated using domain-specific word embeddings

		positive class		negative class	
		manual	automatic	in-class	out-class
FastText-spec	SKIPGRAM	92.76	94.10	75.37	47.82
Word2Vec-spec		92.78	93.82	75.02	47.13
FastText-spec	CBOW	85.59	88.20	51.01	18.74
Word2Vec-spec		85.24	87.88	58.79	23.34
Glove-spec	–	90.76	89.48	66.36	39.83

Using domain-specific word embeddings we can see that the separation between positive and negative classes is very clear, with the pairs of titles from `manual` and `automatic` classes far from the other classes (by at least 18 points). All domain-specific models were able to differentiate well between positive (`manual` and `automatic`) and negative (`in-class` and `out-class`) classes. The **FastText model trained using CBOW** (`FastText-spec CBOW`) was the one with the largest margin between positive and `in-class` (about 34 points) product titles. The same model was also the one with the best largest distance between `in-class` and `out-class` (about 32 points) product titles.

Figure 2a shows the cosine distance values distribution, in each class, generated by `FastText-spec CBOW` model. From these values it is possible to set a threshold for similar products as, for instance, those with cosine similarity above 80.



(a) `FastText-spec CBOW` **(b) `Glove-NILC`** **(c) `mBERT`**

Figure 2. Cosine similarity values distribution, in each class, generated by the best models in each category: domain-specific (a), general-domain (b) and BERT (c).

In our second experiment, we evaluated the performance of the general-domain word embeddings in the same task, obtaining the average cosine similarity values presented in Table 3.

As expected, the average cosine similarity values calculated using general-domain word embeddings were lower than those calculated using domain-specific word embeddings. In this case, the model that best separated the classes was the `Glove-NILC` (with

Table 3. Average values of cosine similarity calculated using general-domain word embeddings

		positive class		negative class	
		manual	automatic	in-class	out-class
FastText-NILC	SKIPGRAM	88.80	89.72	66.90	55.86
Word2Vec-NILC		81.76	81.63	45.15	29.56
FastText-NILC	CBOW	83.42	82.93	50.52	38.62
Word2VeC-NILC		79.11	79.97	40.37	25.70
Glove-NILC	–	80.46	79.31	40.50	23.14

about 39 points between positive and in-class). However, all general-domain models were not so good in distinguishing in-class from out-class: Glove-NILC separating them by only 17 points. This fuzzy boundary between in and out-class products is easily observed in Figure 2b. Furthermore, in this case a threshold of 80 for similar products would label many of those products in automatic class as non-similar ones. Thus, the domain-specific word embeddings perform better than the general ones in calculating product title similarity.

Finally, we also evaluated how well the general-domain pre-trained BERT models – mBERT, BERTimbau Base and BERTimbau Large – can distinguish between title products from the four classes of similarity.

Table 4. Average values of cosine similarity calculated using BERT models

	positive class		negative class	
	manual	automatic	in-class	out-class
mBERT	86.01	85.35	62.93	54.51
BERTimbau Base	89.28	90.45	70.96	59.27
BERTimbau Large	93.88	94.81	85.57	77.60

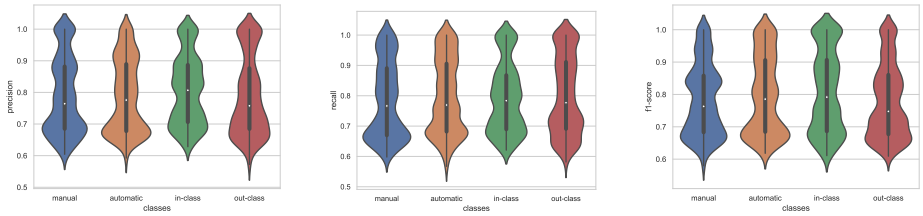
It can be noticed from the values in Table 4, that **mBERT** was the BERT model which best separated between positive and negative classes (with about 22 points between them).

From Figure 2 it is possible to notice that mBERT seems to be the best choice for separating between positive and negative classes. This insight is confirmed when we take a look at the numbers. For example, if we set a threshold of 80 for the cosine similarity, the number of instances incorrectly classified as similar are: 13 for FastText-spec CBOW, 5 for Glove-NILC and only 1 for mBERT. With the same threshold, the number of instances incorrectly classified as non-similar are: 55 for FastText-spec CBOW, 78 for Glove-NILC and 56 for mBERT. So, the best model for calculating product title similarity was the **general-domain pre-trained mBERT model**.

We also investigated if BERTScore [Zhang et al. 2020] calculated using mBERT would lead to better results. However, as can be noticed from the values in Table 5, the BERTScore calculated using mBERT was not so good as the other models in separating the similar products from those not similar. The inadequacy of BERTScore for this task is easily noticed when we analyse the graphs in Figure 3, where it is impossible to clearly separate the classes.

Table 5. Average values for BERTScore calculated using general-domain `mBERT` model

	positive class		negative class	
	manual	automatic	in-class	out-class
Precision	89.02	89.48	71.96	66.12
Recall	88.71	88.59	72.54	66.64
F1	88.80	88.99	72.20	66.35



(a) `mBERT`'s precision **(b) `mBERT`'s recall** **(c) `mBERT`'s F1**

Figure 3. BERTScore values distribution, in each class, generated by `mBERT`.

5. Conclusions and Future Work

From the results of the experiments presented in this paper, we can conclude that domain-specific word embeddings are effective in measuring the similarity between product titles. Among the domain-specific models we trained, the FastText with CBOW showed the best results. However, the best approach for distinguishing between similar and non-similar products was calculating the cosine similarity using the multilingual pre-trained general-domain BERT model.

As future work we intend to fine-tune the Brazilian Portuguese BERTimbau model [Souza et al. 2020] for our task and measure how well a domain-specific fine-tuned BERT model, for Portuguese, would perform in calculating product title similarity. Another proposal for future work is to expand our product title similarity task by including image processing techniques in order to develop a multimodal system.

Finally, although the experiments present in this paper were carried out for Brazilian Portuguese, the product title similarity measuring approach evaluated here is language independent and can be easily replicated for other idioms.

Acknowledgments

This paper and the research behind it would not have been possible without the support of americanas s.a. Digital Lab, specially José Pizani, Ester Campos and Allan Batista, who closely followed this research. This work is is part of the project "Dos dados ao conhecimento: extração e representação de informação no domínio do e-commerce" (Projeto de extensão - UFSCar #23112.000186/2020-97).

References

Alam, F., Afzal, M., and Malik, K. M. (2020). Comparative analysis of semantic similarity techniques for medical text. In *2020 International Conference on Information Networking (ICOIN)*, pages 106–109.

Arts, S., Cassiman, B., and Gomez, J. C. (2017). Text matching to measure patent similarity. *Strategic Management Journal*, 39.

Aryal, S., Ting, K. M., Washio, T., and Haffari, G. (2019). A new simple and effective measure for bag-of-word inter-document similarity measurement. *arXiv preprint arXiv:1902.03402*.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Hartmann, N. S., Fonseca, E. R., Shulby, C. D., Treviso, M. V., Rodrigues, J. S., and Aluísio, S. M. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In *Proceedings of Symposium in Information and Human Language Technology*, pages 122–131, Uberlândia, MG, Brasil. SBC.

Lo, C.-k. (2017). MEANT 2.0: Accurate semantic MT evaluation for any output language. In *Proceedings of the Second Conference on Machine Translation*, pages 589–597, Copenhagen, Denmark. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Rodrigues, E. L., Fernandes, L. A., Rodrigues, E. F., de Arruda, I. P., and Moia, R. P. (2014). A importância da distribuição no comércio eletrônico. *INOVAE-Journal of Engineering, Architecture and Technology Innovation (ISSN 2357-7797)*, 1(1):24–38.

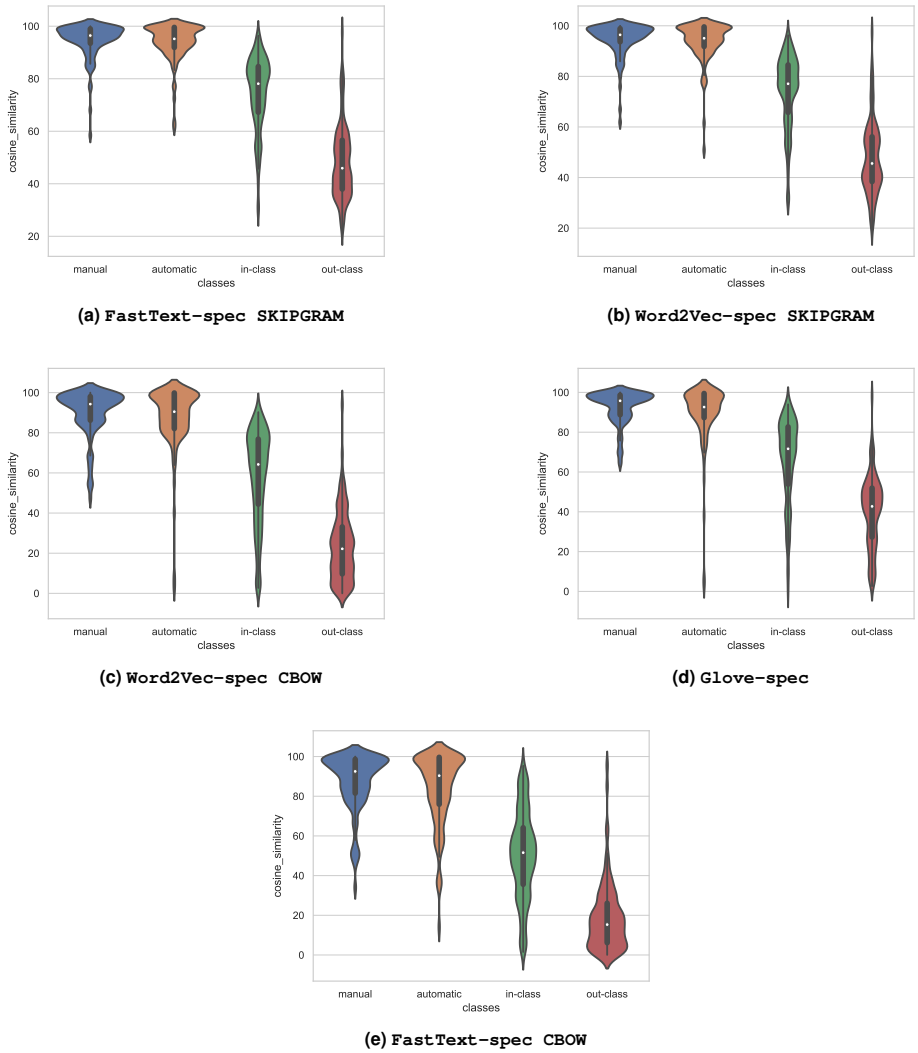
Rosa da Silva, R., Fernandes, E., Motta, E., Akira, E., Guarino, R., and Alvim, L. (2017). Offer categorization for price comparison websites: Word embedding approaches. In Martí, L. and Sánchez Pi, N., editors, *Anais do 13 Congresso Brasileiro de Inteligência Computacional*, pages 1–12, Curitiba, PR. ABRICOM.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: Pretrained BERT Models for Brazilian Portuguese. In Cerri, R. and Prati, R. C., editors, *Lecture Notes in Computer Science*, volume 12319, pages 403–417, Cham. Springer International Publishing.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). BERTScore: Evaluating Text Generation with BERT. In *Proceedings of the International Conference on Learning Representations*.

A. Violin plots

In this Appendix we group all the violin plots for the cosine similarity values calculated using all the domain-specific and general-domain word embeddings.



(a) `FastText-spec SKIPGRAM`

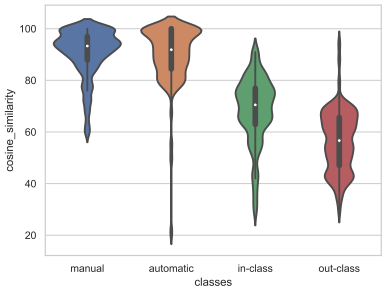
(b) `Word2Vec-spec SKIPGRAM`

(c) `Word2Vec-spec CBOW`

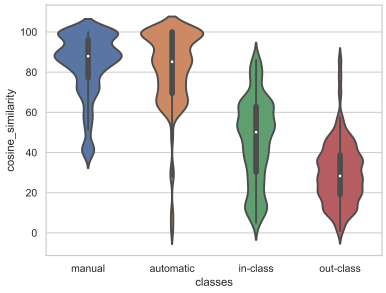
(d) `Glove-spec`

(e) `FastText-spec CBOW`

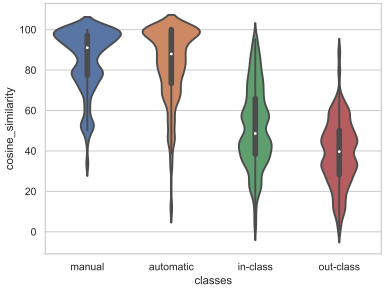
Figure 4. Cosine similarity values distribution, in each class, generated by domain-specific word embeddings.



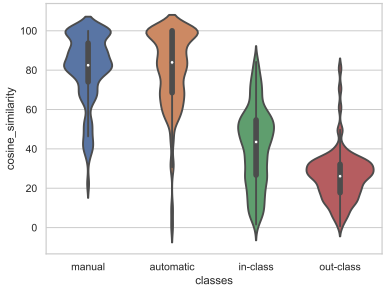
(a) `FastText-NILC SKIPGRAM`



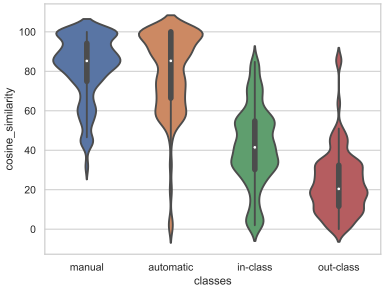
(b) `Word2Vec-NILC SKIPGRAM`



(c) `FastText-NILC CBOW`

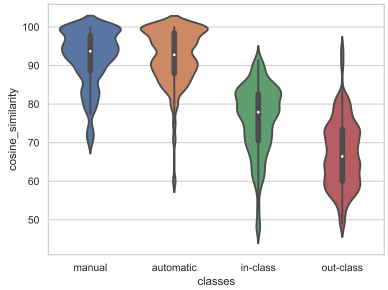


(d) `Word2Vec-NILC CBOW`

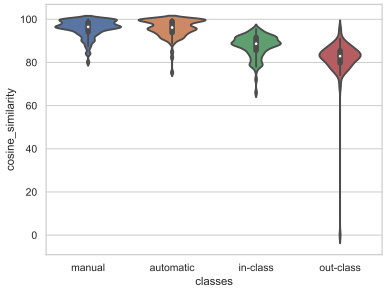


(e) `Glove-NILC`

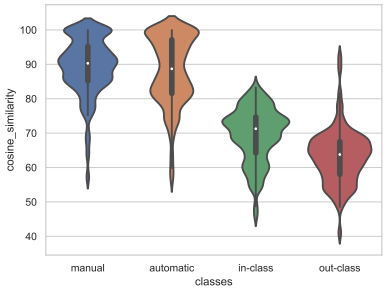
Figure 5. Cosine similarity values distribution, in each class, generated by general-domain word embeddings.



(a) `BERTimbau Base`



(b) `BERTimbau Large`



(c) `mBERT`

Figure 6. Cosine similarity values distribution, in each class, generated by general-domain BERT models.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Augmenting Customer Support with an NLP-based Receptionist

André Barbosa[2], **Alan Godoy**[1]

[1] QuintoAndar, São Paulo, Brazil

[2]University of São Paulo, São Paulo, Brazil

andre.barbosa@ime.usp.br

alan.godoy@quintoandar.com.br

Abstract. *In this paper, we show how a Portuguese BERT model can be combined with structured data in order to deploy a chatbot based on a finite state machine to create a conversational AI system that helps a real-estate company to predict its client's contact motivation. The model achieves human level results in a dataset that containts 235 unbalanced labels. Then, we also show its benefits considering the business impact comparing it against classical NLP methods.*

1. Introduction

Effective customer support is an essential piece for any service company. On the one hand, as a business grows, it can hire more agents and create specialized departments focusing on each possible problem a customer may face, increasing the availability of support for clients. On the other hand, growth brings new challenges: knowledge about client issues gets scattered across the company, information regarding processes may not diffuse properly to all agents, clients may have trouble finding the specific department able to help them, to name few. As a real estate company, QuintoAndar[1] has to deal with a wide range of issues involving individuals as distinct as tenants (and prospective tenants), landlords, agents (realtors, inspectors, photographers, etc) and building administrators. The channel of choice for most contacts is WhatsApp, a messaging app widely adopted in Brazil and other countries. This introduces even more complexity in terms of user identification and routing, as any WhatsApp user can start a conversation with our customer support team simply by sending an open text message.

To improve the routing quality, speed, and automate the collection of all necessary information before a human analyst is allocated to the ticket, our team developed a chatbot system that acts as a receptionist for customer support. This chatbot (presented in section 4) uses a state-machine dialog manager to integrate multiple machine learning (ML) based classifiers and business rules to define the messages to be sent and actions to be taken. The implementation of such service brought a lot of positive results to QuintoAndar, allowing the full automation of chat triage in customer support, a task performed previously by a dedicated team, which was reallocated to other departments.

Beyond presenting a chatbot architecture that combines multiple ML models and discussing how we merged textual information with structured data in order to better

[1]A quick presentation about QuintoAndar and its business model can be found here.

understand an user's context, the main contribution of this work is to bring data about the results achieved by state-of-the-art models in a real-world application. Much has been discussed about the fantastic results achieved by recent Natural Language Processing (NLP) models for the English language both in academic literature an in industry. How these outcomes transfer to Brazilian Portuguese is not as clear though. We hope that, by sharing our experience, we may shed some light on how Portuguese-speaking companies and research groups may use these technologies outside common benchmark problems.

In summary, the main goal of this paper is to show how we evolve until we achieve human-level results in a real-world application through combining Portuguese BERT with structured metadata from our clients with a complete end-to-end system.

2. Related Work

Customer support is a natural candidate to be one of the areas most benefited by the recent boost in NLP research and application [Wan and Chen 2018, Liu et al. 2020, Molino et al. 2018]. By having a proper system for customer triage, a company can match its user to the agent most likely to solve their problem and also provide the agent with contextual information — e.g.: selected user information, procedure suggestion and a list of similar issues — to make their work more effective. Alexandra DeLucia and Elisabeth Moore [DeLucia and Moore 2020] used IT support tickets in a high-performance computing laboratory to study automatic categorization and similar ticket recommendation. They combined classical NLP pre-processing, including steps as stop-word removal, stemming and topic modeling, with a random-forest classifier to select the most appropriate label among 93 possible categories and retrieve other similar tickets. A similar work was done by Fotso *et al.* [Fotso et al. 2018], that created a system to classify client emails in 12 possible categories, using it to select relevant articles to be automatically sent as response. The team combined textual information with data regarding the user's prior interactions with their products, using word-embeddings, a BiLSTM network and an attention-based mechanism to predict the most adequate class.

Rather than applying a single machine learning model to classify the user demand, it's also possible to create a dialog system that interacts with the customer to try solving their problem or to ensure that all relevant information was provided before assigning the ticket to an agent. Such system demands not only a model that extracts information from user messages but also a dialog management module to track the current dialog state and define which action to perform in each moment and also a module to generate the messages sent to the user [Dai et al. 2020, Jurafsky and Martin 2021, Zhang et al. 2020].

When considering Brazilian Portuguese, however, there are few works reporting applications of NLP in industrial applications. Finardi *et al.* [Finardi et al. 2021] built a BERT language model [Devlin et al. 2019] to be used for customer support in a large Brazilian bank, evaluating its performance for sentiment analysis, question answering and named entity recognition (NER) using datasets extracted from user interactions with the bank's chatbot. Azevedo *et al.* [Azevedo et al. 2020] created a system to automatically route customer emails to four different boxes using a Support Vector Machine (SVM). Works have also been published regarding applications in law, as [Bonifacio et al. 2020] — which extensively studies the impacts of fine-tuning of transformer-based language models using legal texts in NER tasks — and [Dal Pont et al. 2020] — that provides an

empirical study of word embeddings in legal domain.

On the academic front, however, many works were produced recently related to dialog systems that can be applied for customer support in Portuguese. Santos *et al.* [Santos et al. 2020] and Melo and Coheur [Melo and Coheur 2020], for instance, built retrieval-based conversational agents trained to answer specific data. Carvalho *et al.* [Carvalho et al. 2020], on the other hand, created an LSTM-based common-sense module to augment interactions in a dialog system. A recent significant contribution was also made by Souza *et al.* [Souza et al. 2020] that made open-source a Brazilian Portuguese pre-trained BERT language model, showing how such mono-lingual model was able to surpass the performance achieved by a multi-lingual BERT for named entity recognition (NER).

3. Problem Definition and Challenges

A typical customer support flow at QuintoAndar (Fig. 1) starts with a customer contacting our team through the channel of their choice — namely, phone, WhatsApp or email/support form. Whichever channel is chosen, the user is required to state what kind of support is needed, an information that is used to define to which department the contact should be directed. After the department is defined, a task allocation system selects among all online agents who should be responsible for handling the customer's issue. The role of such agent is to comprehend the user's demand and get all relevant information to solve it — if it is simple enough, it can be resolved immediately; if it requires more complex actions, a task is created for the appropriate back-office team.

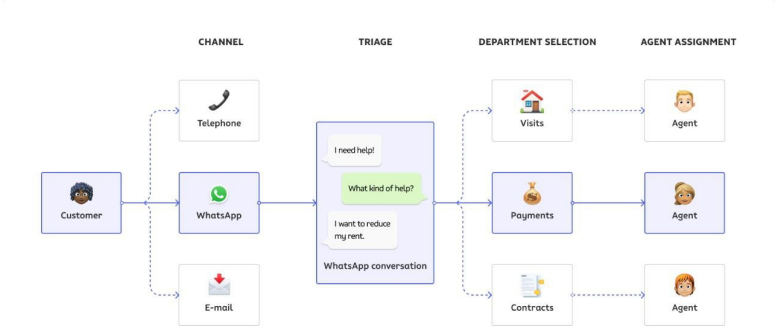


Figure 1. A typical flow of customer support at QuintoAndar.

In a digital real-state company, however, there are hundreds of different types of contacts, going from simple questions about using our site to search for a house to requests for intermediation of complex tenant-owner issues like negotiating a temporary rent reduction. This means that matching a user to the right specialist is easier said than done. On the one hand, presenting a menu with a multitude of customer support departments may be time-consuming and ineffective, as users may not be familiar with each department's responsibilities and are likely to make mistakes when choosing an option. On the other hand, letting the users freely declare what they need has its own drawbacks, such as the possibility that the user's explanation does not have sufficient context regard-

ing their demand and the issue of how to convert that explanation into a decision about the destination department of the contact.

For WhatsApp — our main support channel and the focus of the present work — this issue gets trickier, though: the app is ubiquitously used in Brazil as a means of communication with family and friends. Added to this is the fact that Brazilian clients are used to small real estate companies in which commercial interactions are based on a person-to-person relationship. Contacts, thus, may start with a simple *"Hi!"*, with a complete account of the user's issue or with a partial description that refers to past contacts.

Prior to this work, triage and department selection was carried out by a team dedicated solely to verifying that sufficient context was provided, asking users to give more information if necessary, and directing each chat to the appropriate department. Of course, this was a far from ideal situation. A large number of skilled agents were diverted from customer troubleshooting to perform a highly repetitive task that added latency to proper chat routing and stiffened any processes changes that incur in changes in departmental responsibilities, as any such changes would require retraining the triage team.

To sustain company's scalability, our team developed a "receptionist" chatbot that should be able to:

1. Determine whether the user has provided enough information in their initial message, requesting a better description if deemed necessary.
2. Combine this textual information with data about the user's relationship with QuintoAndar to predict the reason for the current contact.
3. Automate the collection of basic data, such as information that helps to identify users who could not be identified based on their phone number and data to help the agent verify the authenticity of the user.
4. Use business rules together with the predicted contact reason to determine to which department the chat should be directed.
5. Provide the selected agent with the predicted reason for the contact in order facilitate access to the appropriate guidelines for that service and speed up the mandatory annotation after the conversation is finished.

This was not without its hurdles. First, there were more than 300 standardized possible contact reasons with a substantial gray area between them, ensuring a great deal of noise in labels. Classes were highly unbalanced, with some containing only a handful of chats in a given year, while others classes totaling thousands in a single week. Finally, as a Portuguese-speaking company, there were no open resources available related to customer service, so we could rely only on pre-trained models and our own private datasets to develop the chatbot.

4. System Architecture

Following the dialog-state architecture [Jurafsky and Martin 2021], the bot engine we created has the following basic components:

- Dialog memory: a flexible memory to store all relevant information produced by handlers during the conversation, including special field $\mathcal{S}$ that indicates the automaton's current state.

- Handlers: processors used by the chatbot to extract relevant information from input messages or make decisions about what to do next.
- Finite-state automaton: a recipe that indicates which handler should be called at each state $\mathcal{S}$ and how to use a policy handler results to select the next state.

Handlers, in turn, can be of two different types:

- Message processing handlers: responsible for extracting information from input messages and storing it on dialog memory. The present work focuses on proposing two machine learning models that act as message processing handlers.
- Policy decision handlers: responsible for deciding what the system should do given the information stored in dialog memory. Examples of possible actions are performing some calculations and storing their results in the memory, sending a response to the user or transferring the chat to a certain department. When a policy handler decides to send the user a response, it forwards asks the language generation unit to select a template to use to create the message.

Each handler has its own implementation, which may range from simple application of business rules to complex ML models and third-party services. This work makes use of two different NLP-based handlers that will be explained on the next subsections.

4.1. Context evaluation model

A common behavior by our users was to start the conversation by saying a greeting, only stating what she needed after being asked by the support agent. This posed a significant challenge for our receptionist bot, as it would predict contact reasons and select destination departments based on messages that were not at all explanatory. A simple rule of thumb to avoid this problem would be to always discard the first message in a conversation. However, this would hurt the client's experience, which is only worsened by the fact that many customer support contacts are made by already stressed users.

To alleviate this issue, we created a model to predict whether or not a particular message was sufficiently explanatory. Based on such prediction our chatbot may decide to ask the client for more input. A dataset with 5348 chats randomly sampled from QuintoAndar's ticketing system was manually annotated by two customer support agents. They could classify a message as one of the following: *has context*, *no context*, *returning client* and *low value* (a message that is not intended to create a new conversation, as *"Thanks!"*). The final dataset contained 2926 chats labeled as its content has enough context and 2422 labeled as not enough context (*no context* and *low value*; *returning client* messages were discarded), endorsing the relevance of this first model in an efficient reception.

Our goal was to have a simple model that would introduce only a small delay in response time, so we decided to user bag-of-words (BoW) representation with a logistic regression (LR) as classifier. Text pre-processing included accent stripping, uncasing and stop-word removal; no stemming or lemmatization was used. Bag-of-words representation was then computed for 3-grams with a vocabulary of 5000 items. LR hyperparameters (C, penalty type and class weight) were selected through Bayesian optimization — the solver used was 'lbfgs' [Zhu et al. 2011]. The resulting model achieved accuracy of 85% on the testing split, a value considered good enough for production as most errors in this step could be fixed by human agents after chat allocation.

4.2. Contact reason prediction model

The main model for the chatbot is the contact reason prediction model. As explained above, its main purpose is to combine the user's message with tabular information indicating their relationship with QuintoAndar to predict how likely the contact is to be related to each of 306 possible standardized contact reasons (see Table 1 for some examples).

`cr_pg`	Payments of real-estate brokers.
`ft_ag_alteracao`	Schedule changes for photographers.
`iq_pr_reserva`	Issues related to house reservation by a tenant after a proposal is accepted by an owner.
`pp_cm_venda_imovel`	An owner informs that she intends to sell the house she currently rents.

Table 1. Examples of contact reasons defined by QuintoAndar's process team.

The association of text and tabular data is important as a way to avoid requiring users to explain their whole history to the chatbot. Consider the phrase *"I need to cancel the visit tomorrow."*. It is a complete yet very ambiguous message: is it related to a photographer that wants to reschedule a photo shoot, to a potential tenant who is not interested on visiting an apartment anymore, or to a real estate agent that will not be able to present some house and wants to leave it to another colleague? Only by having access to the user's relationship with QuintoAndar the model would be able to accurately predict the contact reason without requesting further context. Therefore, we used 66 handcrafted features available from our feature store [Marques 2020, Hermann and Balso 2017]. Some examples are the type of the last automatic message sent to the user (and time since it was sent), the contact reason for the last ticket (and time since it was created), whether the user is a registered agent, number of rented houses (as owner) and number of ended contracts (as tenant). These features are pre-processed, performing one-hot encoding on all categorical columns and scaling the numeric ones to have zero mean and unit variance.

We treat both feature groups (textual and tabular) as different modules that are fed into a separate classifier. In a first version (V1 - bag-of-words), we used a simple unigram bag-of-words to extract features from messages. After analyzing this model errors, we noticed that, despite showing good results, it had trouble with synonyms and complex sentences. In a second version of the model (V2 - BERT) we addressed this issue by using for textual feature extraction a representation extracted by a Portuguese version of a BERT model [Devlin et al. 2019, Souza et al. 2020].

5. Experiments

We briefly discuss the results achieved for version 1 and version 2 of contact reason prediction model, taking into consideration different setups as well as analyzing overall model's performance.

5.1. Dataset

The full dataset used to train the contact reason model contains data for 639159 chats between May 2019 and August 2020 manually annotated by support agents, selecting only user messages sent before an agent entered the conversation. We used out-of-time

split to partition the between training (511327 chats), validation (63916 chats) and testing (63916 chats) sets. With respect to classes, originally the data contained 306 distinct ones — after filtering all tickets belonging to classes with less than 50 samples on training set the final dataset contained 235 different labels. To fine-tune BERT on our data, since it is a considerably heavy model, we have used a smaller portion of dataset, with 178578 samples for training, 19843 for validation and 66141 for testing.

5.2. Metrics

To analyze the model performance, we have calculated top-1 and top-3 test accuracy both for contact reason and for department[2].

5.3. V1 - Bag-of-words

For this first solution approach, we have evaluated three possible classifiers: logistic regression (LR), random forest (RF) and multilayer perceptron classifier (MLP). Hyperparameters were selected through Bayesian optimization with 100 evaluated configurations [Biewald 2020]. Classifiers comparison are presented in Table 2.

Table 2. Test accuracy for contact reason and department. Higher is better.

Model	Top-1 accuracy	Top-3 accuracy	Top-1 dept. accuracy	Top-3 dept. accuracy
LR	42.8%	63.6%	77.8%	84.6%
RF	40.7%	61.2%	75.2%	82.7%
MLP	**44.1%**	**65.1%**	**78.2%**	**85.0%**

5.3.1. V2 - BERT

As observed in Table 2, the MLP model obtained the best results according to all metrics. Given the high cost of using a large Transformer-based network and the fact this new version is very similar to V1, we decided to keep the same architecture.

The first step we took was to fine-tune the BERT-Large model made available by Souza *et al.* [Souza et al. 2020] to predict contact reason. Sentences with more than 64 tokens were truncated. Once this fine-tuned procedure was completed, we tested different methods in order to replace the bag-of-words of V1 with a BERT module:

- Use the logits of BERT classification head.
- Use the last layer of BERT language model head as suggested by Devlin *et al.* [Devlin et al. 2019] .
- Concatenate last four layers of BERT language model head as suggested by Devlin *et al.* [Devlin et al. 2019]

The results are shown in Table 3. First of all, it is relevant to notice the large gains achieved by combining both textual and tabular features. Also, we can see that using BERT as textual feature extractor provided better results than using bag-of-words. Considering engineering constraints we have decided to use the logits of BERT classification head + tabular data as version 2 of the contact reason model.

[2]Department selection is performed by summing predicted probabilities for all contact reasons associated to a given department. The department with highest score is selected.

Table 3. Comparison of contact reason accuracy for multiple models.

Model	Top-1 accuracy
BoW alone	38.14%
BoW + tabular data (V1)	44.11%
BERT classifier	45.57%
BERT classifier logit heads + tabular data	53.10%
Last layer LM BERT + tabular data	**53.20%**
Last 4 layers LM BERT concat + tabular data	53.05%

6. Business impacts

To assess the results in production, we have collected data from a triage human team and for a set of heuristics rules that routed simply based on the last automatic message sent to the user (e.g., if the message was related to a visit, then the user was directed to the visits department). The business metric that we decided to follow to compare was the transference rate — *i.e.*, the rate in which a chat already routed to a given support department should be transferred to another department.

We have run tests in production comparing human triage and both chatbot versions. As a security measure to avoid deterioration of user experience, we only routed automatically the 80% of tickets with highest department score leaving all low-confidence tickets to human triage. As the results were good enough (i.e. similar to human performance), we expanded it to 100% to out clients base. Results are presented in Table 4.

Table 4. Production results. Data for human triage and heuristic rules refer were collected prior the tests. Lower the better for both columns.

Model	Transf. rate	Avg. msg. per ticket
Human triage	12.8%	18.2
Heuristics rules	18.3%	11.2
V1 - Bag-of-words (80%)	13.9%	13.2
V2 - BERT (80%)	10.3%	13.7
V2 - BERT (100%)	13.2%	14.2

Considering these results, we can easily say that the BERT embedding combined with tabular data achieved human-level performance. The number of message exchanges until the conversation was ended (avg. message per ticket) was also substantially smaller for clients routed by the chatbot than for those routed by humans. An hypothesis regarding the reduction in message numbers is the fact that by using tabular features the model has access to a large amount of information not easily consumed by humans.

Such results in a real-world application using business metrics endorses the potential of modern NLP techniques for Brazilian Portuguese. We hope that results like these help companies and governments to see that it's now feasible to go much beyond the widespread rigid conversational interfaces based on buttons or simple keywords.

Acknowledgments

We would like to thank Muriel Dias for producing the images in our paper, as well as Marco Antonio Rocha Vinha for reviewing the system architecture shown in section 4.

References

Azevedo, R. F. d., Rodrigues Pereira de Araujo, R., Guimarães Araújo, R., Moreira Bittencourt, R., Ferreira Alves da Silva, R., de Melo Vaz Nogueira, G., Marques Franca, T., Otharan Nunes, J., Ralff da Silva, K., and Regiane Cunha de Oliveira, E. (2020). Screening of email box in portuguese with svm at banco do brasil. In Quaresma, P., Vieira, R., Aluísio, S., Moniz, H., Batista, F., and Gonçalves, T., editors, *Computational Processing of the Portuguese Language*, pages 153–163, Cham. Springer International Publishing.

Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb.com.

Bonifacio, L. H., Vilela, P. A., Lobato, G. R., and Fernandes, E. R. (2020). A study on the impact of intradomain finetuning of deep language models for legal named entity recognition in portuguese. In Cerri, R. and Prati, R. C., editors, *Intelligent Systems*, pages 648–662, Cham. Springer International Publishing.

Carvalho, C. S., Pinheiro, V. C., and Freire, L. (2020). A deep learning model of common sense knowledge for augmenting natural language processing tasks in portuguese language. In Quaresma, P., Vieira, R., Aluísio, S., Moniz, H., Batista, F., and Gonçalves, T., editors, *Computational Processing of the Portuguese Language*, pages 303–312, Cham. Springer International Publishing.

Dai, Y., Yu, H., Jiang, Y., Tang, C., Li, Y., and Sun, J. (2020). A survey on dialog management: Recent advances and challenges.

Dal Pont, T. R., Sabo, I. C., Hübner, J. F., and Rover, A. J. (2020). Impact of text specificity and size on word embeddings performance: An empirical evaluation in brazilian legal domain. In Cerri, R. and Prati, R. C., editors, *Intelligent Systems*, pages 521–535, Cham. Springer International Publishing.

DeLucia, A. and Moore, E. (2020). Analyzing hpc support tickets: Experience and recommendations. *ArXiv*, abs/2010.04321.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding.

Finardi, P., Viegas, J. D., Ferreira, G. T., Mansano, A. F., and Caridá, V. F. (2021). Bertaú: Itaú bert for digital customer service.

Fotso, S., Spanoudes, P., Ponedel, B. C., Reynoso, B., and Ko, J. (2018). Attention fusion networks: Combining behavior and e-mail content to improve customer support. *ArXiv*, abs/1811.03169.

Hermann, J. and Balso, M. D. (2017). Meet michelangelo: Uber's machine learning platform.

Jurafsky, D. and Martin, J. H. (2021). *Speech and Language Processing (3rd Edition)*. Prentice-Hall, Inc., USA.

Liu, C., Jiang, J., Xiong, C., Yang, Y., and Ye, J. (2020). Towards building an intelligent chatbot for customer service: Learning to respond at the appropriate time. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Dis-*

covery & Data Mining, KDD 20, page 3377–3385, New York, NY, USA. Association for Computing Machinery.

Marques, A. (2020). Butterfree: A spark-based framework for feature store building.

Melo, G. and Coheur, L. (2020). Towards a conversational agent with "character". In Quaresma, P., Vieira, R., Aluísio, S., Moniz, H., Batista, F., and Gonçalves, T., editors, *Computational Processing of the Portuguese Language*, pages 420–424, Cham. Springer International Publishing.

Molino, P., Zheng, H., and Wang, Y.-C. (2018). COTA: Improving the speed and accuracy of customer support through ranking and deep networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD 18, page 586–595, New York, NY, USA. Association for Computing Machinery.

Santos, J., Alves, A., and Gonçalo Oliveira, H. (2020). Leveraging on semantic textual similarity for developing a portuguese dialogue system. In Quaresma, P., Vieira, R., Aluísio, S., Moniz, H., Batista, F., and Gonçalves, T., editors, *Computational Processing of the Portuguese Language*, pages 131–142, Cham. Springer International Publishing.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23*.

Wan, M. and Chen, X. (2018). Beyond "how may i help you?": Assisting customer service agents with proactive responses.

Zhang, Z., Takanobu, R., Zhu, Q., Huang, M., and Zhu, X. (2020). Recent advances and challenges in task-oriented dialog system.

Zhu, C., Byrd, R., Nocedal, J., and Morales, J. L. (2011). L-BFGS-B — software for large-scale bound-constrained optimization.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Audio MFCC-gram Transformers for respiratory insufficiency detection in COVID-19

Marcelo Matheus Gauy[1], **Marcelo Finger**[1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (USP)

Abstract. *This work explores speech as a biomarker and investigates the detection of respiratory insufficiency (RI) by analyzing speech samples. Previous work [Casanova et al. 2021] constructed a dataset of respiratory insufficiency COVID-19 patient utterances and analyzed it by means of a convolutional neural network achieving an accuracy of* 87.04%*, validating the hypothesis that one can detect RI through speech. Here, we study how Transformer neural network architectures can improve the performance on RI detection. This approach enables construction of an acoustic model. By choosing the correct pretraining technique, we generate a self-supervised acoustic model, leading to improved performance (*96.53%*) of Transformers for RI detection.*

1. Introduction

COVID-19 is the cause of a major pandemic that threatens to collapse the healthcare systems in many regions of the world. Respiratory insuficiency (RI) is one of COVID-19 symptoms, which often requires hospitalization and is aggravated by a common COVID-19 condition called *silent hipoxia*, low blood oxygen concentration without breath shortness [Tobin et al. 2020]. This work aims to help deal with the COVID-19 pandemic by providing an automated system, based on deep learning techniques, capable of detecting RI in COVID-19 patients. Such an automated system could, for example, support cellphone-based patient triage procedures alleviating the burden on health personnel.

We explore the view of *speech as a biomarker*, by building upon a recently shown fact: it is possible to detect respiratory insufficiency through analyzing spoken utterances in real-life conditions (typically a moderately large sentence). This hypothesis has been previously verified [Casanova et al. 2021] by using a CNN-based deep neural network. This CNN received a moderately large sentence spoken in real life conditions and had to predict whether it came from a patient with RI or from the control group. In this work, we aim to further analyze that hypothesis by studying other network architectures (namely, Transformers [Vaswani et al. 2017]), in an attempt to improve the results previously obtained in [Casanova et al. 2021], with a view of extending it in the future to RI originated from other causes, such as influenza, heart disease or mental illness.

In this work we find that Transformers can be used for detecting respiratory insufficiency with an accuracy of 96.38% up from 87.04% in [Casanova et al. 2021]. To reach that level of performance, we feed the Transformers with a sequence of Mel Frequency Cepstral Coefficients (MFCC) obtained from the patients' audios (henceforth called MFCC-gram Transformers). Like CNN-based detection from [Casanova et al. 2021], the Transformer performance drops significantly (to 82.87%) if we feed it standard spectrogram coefficients (called Spectrogram Transformers after [Gong et al. 2021]).

The Transformers [Vaswani et al. 2017] were shown to be very effective when divided in two parts [Devlin et al. 2018]. The *pretraining phase* generates a language-based acoustic model with unsupervised (or self-supervised) learning by optimizing a generic language prediction task with a large amount of generic data. Then, the acoustic model undergoes a task-specific *refinement phase* in which both the acoustic model and additional task-specific neural modules are trained on smaller-size application data. A *baseline transformer* is one in which pretraining is a random assignment of weights.

Here, we find that MFCC-gram Transformers benefit from being pretrained with large quantities of spoken Brazilian Portuguese audios, which is later refined for the target task of detecting respiratory insufficiency. For pretraining, we explore three known techniques from the literature [Liu et al. 2020b, Liu et al. 2020a] and find that they generally lead to some performance improvement over baseline transformers. Performance reaches 96.53% using the best of the available techniques.

2. Related Work

In addition to [Casanova et al. 2021] there have been other works [Pinkas et al. 2020, Laguarta et al. 2020] which study COVID-19 with deep learning using voice related data. [Pinkas et al. 2020] attempt to detect SARS-COV-2 (the virus that causes COVID-19) from voice audio data, while this work and [Casanova et al. 2021] attempt to detect RI. Furthermore, there have been previous works which support the view of speech as a biomarker [Botelho et al. 2019, Nevler et al. 2019, Robin et al. 2020].

Transformers were designed for NLP [Vaswani et al. 2017, Devlin et al. 2018], and were also later used in audio processing tasks [Liu et al. 2020b, Liu et al. 2020a, Schneider et al. 2019, Baevski et al. 2020, Baevski et al. 2019, Song et al. 2019]. In Mockingjay and Tera [Liu et al. 2020b, Liu et al. 2020a], it was used in phoneme classification and speaker recognition tasks. There it was shown that variants of the Cloze task [Taylor 1953, Devlin et al. 2018] for audio could be used for unsupervised pretraining of Transformers. In Wav2Vec and its variants [Schneider et al. 2019, Baevski et al. 2020, Baevski et al. 2019], a contrastive loss is used to enable unsupervised pretraining, which is later finetuned to speech and phoneme recognition tasks. In Speech-XLNet [Song et al. 2019], a speech based version of the XLNet [Yang et al. 2019] was proposed. The XLNet is a network that maximizes the expected log likelihood of a sequence of words with respect to all possible autoregressive factorization orders.

3. Methodology

3.1. Datasets

For the task of respiratory insufficiency detection, the data used in the refinement phase is the same one used in [Casanova et al. 2021]. There, COVID patient utterances were collected by medical students at COVID wards from patients with blood oxygenation level below 92%, as an indication of RI. Control data was collected by voice donations over the internet without any access to blood oxygenation measurements and were therefore assumed healthy. As COVID wards are noisy locations, an extra collection was made consisting of samples of pure background noise (no voice). This is a crucial step in preventing the network to overfit to the background noise differences in data collection.

The gathered audios contained 3 utterances:

- A long sentence with 31 syllables. It was designed by linguists to be long enough to have reading pauses while being simple for even low literacy donors to speak.
- A widely known nursery rhyme for readers with reading impediments.
- A well known song along the lines of 'Happy birthday to you'.

As suggested in [Casanova et al. 2021], we select only audios from the first utterance and sample balance the dataset by class and sex. The presence of ward background noise in the patient audios is treated in a similar way: we insert noise to the control group as that is easier than removing it from the patients' signal. This prevents that we eliminate from the signal, audio that is relevant to the network's classification.

We employ the same division in training, validation and test as done in [Casanova et al. 2021]. The best signal-noise ratio audios are included in the test set. The second best audios are in the validation set. This is done to detect training overfitting. Table 1 contains information on the number of audio files for each class.

Sets	Control			Patients			Total Audios
	Male	Female	Mean duration(s)	Male	Female	Mean duration(s)	
Training	59	84	8.15	83	66	13.18	292
Validation	8	8	7.75	8	8	10.78	32
Test	22	26	7.77	28	32	9.43	108

Table 1. Filtered dataset information.

For the pretraining phase, we use datasets containing Brazilian Portuguese speech. These datasets are NURC-Recife [Oliviera Jr et al. 2016], ALIP [Gonçalves 2019], C-Oral Brasil [Raso and Mello 2012] and SP2010 [Mendes 2013]. Together, they contain more than 200 hours of Brazilian Portuguese speech.

3.2. Preprocessing

As we face similar audio processing issues as [Casanova et al. 2021], we employ similar preprocessing steps. In the dataset, the majority of audios were sampled at $48kHz$. We preprocess the files using Torchaudio 0.9.0. We extract either the mel-spectrogram (for Spectrogram Transformers) or the MFCCs of the audios with default Torchaudio parameters and retain 128 coefficients. Torchaudio, by default, employs a Fast Fourier Transform [Brigham and Morrow 1967] with a $400ms$ window and hop length 200.

As the dataset has an inherent imbalance in the audio lengths from patients and control we do not use the full audios of the first utterance. Instead, we break each audio into 4 seconds chunks, with a windowing of 1 second steps. Such a windowing method was observed in [Casanova et al. 2021] to be more effective than, for example, padding the audios with zeros to make all the audios have the same length. The windowing technique solves the problem of the imbalance between audio lengths and guarantees the network will not pay too much attention to the audio lengths and instead focuses on the content. The windowing technique also serves as a kind of data augmentation as, for example, an audio with 8 seconds becomes 5 audios with 4 seconds. We observe that the windowing should be done before the spectrogram or MFCCs feature extraction.

3.3. Noise insertion

The noise in COVID wards is a serious bias source. This can be seen in our experiments and in the original work with the dataset by [Casanova et al. 2021]. One potential way of dealing with this bias source is to filter the noise and eliminate it. However, this has the risk that we eliminate important low-energy information from the data, information which would have been useful in detecting whether a patient had RI. Moreover, eliminating the noise could also create extra biases, as different procedures for eliminating patient and control noises would be required. Thus, instead of eliminating the noise, we consider it much easier to insert the noise present in the COVID wards into all the audio samples.

The original dataset contained 16 samples of 1 minute each containing just the background noise present in COVID-19 wards. These noise samples are added to all the training, validation and test audios, similarly to what was done in [Casanova et al. 2021]. We experiment with the amount of noise we add to each of the audio files. During training, audio samples are injected with one or more noise samples. These are selected randomly from the pool of noise samples each time an audio is used for training. The starting point of each noise sample is also selected randomly. Lastly, a factor to change the intensity of the sample is drawn. This factor is limited by a maximum amplitude value which depends on the patient audio noises. This process is similar to the one in [Casanova et al. 2021] and the goal is inserting noise as similar to the pre-existing noise as possible.

3.4. Transformers

We consider two types of Transformers: MFCC-gram Transformers and Spectrogram Transformers. They are equivalent except in the data features that are fed to them: MFCC-gram Transformers receives MFCC audio features and Spectrogram Transformers receive mel spectrogram audio features. Our Transformers are equivalent to the Transformer Encoder units described in [Vaswani et al. 2017]. Namely, we use a multi-layer Transformer encoder (3 layers) with multi-head self-attention. Each encoder layer has two sub-layers, the first being a multi-head self-attention network and the second being a fully connected feed-forward layer. Each sub-layer has a residual connection followed by layer normalization [Ba et al. 2016]. Every encoder layer and sub-layers produce outputs of dimension 512. In addition to the attention sub-layers, each encoder layer contains a fully connected feed-forward network with an inner layer of dimension 2048.

In order to generate the sequence of tokens that is sent to the Transformers the MFCC and/or Spectrogram is split into its frames. Each frame of the MFCC or spectrogram corresponds to one token fed to the sequence. We also attempted joining multiple frames into one token but this typically produced worse results than the one to one framework. We use sinusoidal positional encoding [Vaswani et al. 2017, Liu et al. 2020b, Pham et al. 2019] to make our model position aware. As suggested by [Liu et al. 2020b], each frame is first projected linearly to a hidden state of dimension 512.

Our Transformers are trained in two phases: pretraining and refinement. In the pretraining phase, we leverage the unsupervised training techniques described in Section 3.5 to build an acoustic model over generic audio data. In the refinement phase, the pretrained Transformers is refined over COVID related audio data. For some experiments, we bypass the pretraining phase by initializing the Transformers with a random assignment of weights and refining that over the COVID data. This is done to get a baseline

performance and we call these Transformers the baseline Transformers. We will name our Transformers types baseline MFCC-gram Transformers and baseline Spectrogram Transformers when we consider Transformers which bypass the pretraining phase.

Our code is based on the guide "The annotated Transformer"[1]. While our Transformers are small in comparison to the ones used, e.g. in BERT [Devlin et al. 2018], the amount of available data for respiratory insufficiency detection is also rather small so we do not expect that larger Transformers would yield significantly improved results. Once more data is available, it is recommended to also increase our Transformers.

3.5. Unsupervised pretraining: acoustic model construction

We describe three techniques to pretrain acoustic models in a self-supervised way. They are based off Masked acoustic modelling [Liu et al. 2020b]. This erases a fraction of the input and tries to reconstruct the erased parts from the remaining frames. They are bidirectional methods and the reconstruction depends on both left and right contexts.

Time Alteration: also called Masked acoustic modelling [Liu et al. 2020b]. Start by selecting frames up to 15% of the input[2], 1) mask them all to zero 80% of the time, 2) replace all with a random frame 10% of the time or 3) leave the frames be in the remaining 10% of the time. The goal of this process (as opposed to always masking the frames) is to alleviate the mismatch between training and inference.

Channel Alteration: this techinique is from [Liu et al. 2020a]. Randomly mask a block of consecutive quefrency channels to zero for all time steps of the input sequence. First, the width W_C of the block is selected uniformly from $\{0, 1, \ldots, W\}$ where W is a 10% fraction of the total number of channels. Second, sample a channel index I_C from $\{0, 1, \ldots, H - W_C - 1\}$ where H is the total number of channels in the input. Then, channels from I_C to $I_C + W_C - 1$ are masked to zero. Observe that (as with time alteration), a fraction of the time none of the channels will be masked.

Noise Alteration this technique is from [Liu et al. 2020a]. Apply sampled Gaussian noise to change the magnitude of the inputs with a probability of 10%. For that end, we sample a random magnitude matrix with the same size as the input. Each element in the matrix is sampled from a normal distribution with mean zero and 0.2 variance. The matrix is then added to the real input frames.

4. Results and Discussion

Here we show the results obtained by the two experiments performed: the first where we compare baseline MFCC-gram Transformers, baseline Spectrogram Transformers and the CNN from [Casanova et al. 2021], and the second where we try different unsupervised pretraining techniques to improve baseline Transformers by building an acoustic model.

First, we note that when no ward noise is added to either the patient or control files, baseline MFCC-gram Transformers performs very well (98.89 ± 0.38) in the test files. However, this performance drops dramatically (to 70.07 ± 3.15) if we add noise to the test files and this is a strong sign the model is biased by the noise. This bias is less

[1] `http://nlp.seas.harvard.edu/2018/04/03/attention.html`

[2] More precisely, we select a fraction of the frames in chunks of a certain size so that the total number of frames masked amounts to 15%. In the experiments, the chunk size was 7.

extreme than what was observed at the MFCC-gram CNN in [Casanova et al. 2021] but is still present. Therefore, in our experiments, noise is added to the training and test files.

In the first experiment, we consider baseline Transformers and bypass the pre-training phase. We vary the amount of ward noise we add to the training and test files. We add to the audio files between 0 and 3 noise files, including either the same amount of noise files to the patient and control audio files or one more file to the control files. This is comparable to the Experiments $3.x$ from [Casanova et al. 2021] and we can directly compare baseline MFCC-gram Transformers, baseline Spectrogram Transformers with the CNN from [Casanova et al. 2021]. We perform each experiment for 20 epochs and repeat the experiments 10 times. The batch size is set to 16. The results are in Table 2. We show both the performance when including noise as well as the performance without including noise in the test samples. Figure 1 shows the same data as Table 2.

Model	Noise Samples		Accuracy (with noise in test samples)	Accuracy (without noise in test samples)
	Patient	Control		
Baseline MFCC-gram Transformers	0	1	$\mathbf{96.38 \pm 0.72}$	96.85 ± 0.84
	1	1	96.30 ± 1.12	$\mathbf{97.36 \pm 1.89}$
	1	2	95.39 ± 1.26	96.44 ± 1.72
	2	2	95.68 ± 0.48	97.35 ± 1.01
	2	3	94.33 ± 1.48	96.53 ± 1.13
	3	3	94.86 ± 0.75	96.63 ± 1.09
MFCC-gram CNN	0	1	74.07 ± 1.93	61.11 ± 8.40
	1	1	86.11 ± 2.98	66.67 ± 3.74
	1	2	83.33 ± 3.34	84.26 ± 6.17
	2	2	85.19 ± 0.93	88.89 ± 0.53
	2	3	85.19 ± 1.85	74.07 ± 5.10
	3	3	87.04 ± 0.93	91.67 ± 2.98
Baseline Spectrogram Transformers	0	1	82.87 ± 1.48	68.73 ± 3.88
	1	1	82.84 ± 1.82	82.65 ± 2.25
	1	2	80.75 ± 2.22	77.18 ± 1.56
	2	2	79.02 ± 3.19	82.05 ± 2.57
	2	3	78.07 ± 2.78	74.67 ± 1.99
	3	3	78.73 ± 2.33	81.96 ± 1.97

Table 2. The performance of the Transformers and the CNN is shown in Table 2. The different lines show performance of the network according to the number of noise files added to the test files, both for patients and control.

We observe a significant improvement in performance for baseline MFCC-gram Transformers when compared to the MFCC-gram CNN. When including noise in test samples, the best performance is attained by baseline MFCC-gram Transformers where we add a single noise file to the control files and keep the patient files unchanged. When we compare without noise being added the best performance is attained by baseline MFCC-gram Transformers where we add a single noise file to both the patient and control files. We would like to point out though that the differences are rather small and baseline MFCC-gram Transformers performs well as long as some noise is added.

For the second experiment, we fix the amount of ward noise we insert to the train-

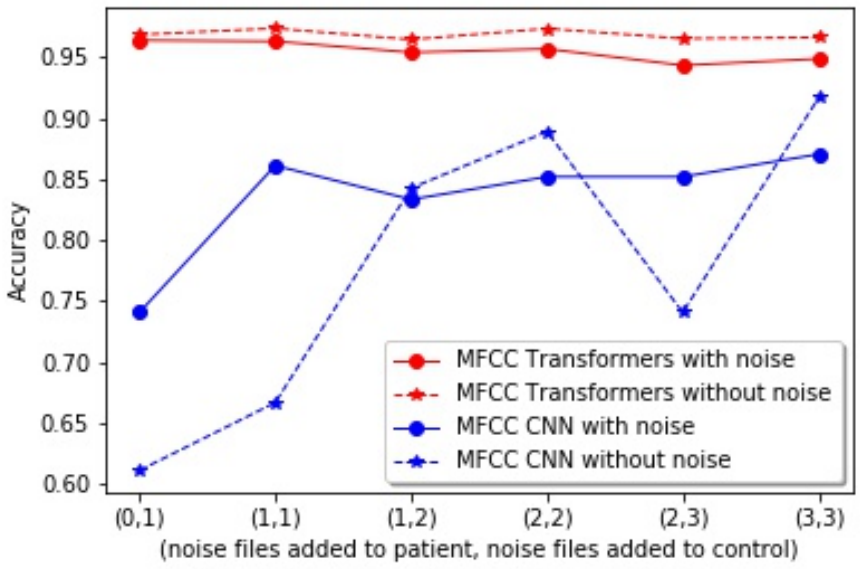


Figure 1. This has the same data as Table 2. The y axis shows accuracy and the x axis shows the number of noise files added to patient and control files.

ing and test files to be a single noise file for both patient and control audio files. We vary the technique employed for unsupervised pretraining, attempting time alteration, channel alteration and noise alteration techniques as described in Section 3.5. We pretrained on the corpuses of NURC-Recife, C-Oral Brasil, SP 2010 and ALIP. Pretraining consisted of 5 epochs on the data of all those corpuses, splitting each file into 4 seconds audio with a 1 second window step. Finetuning on the respiratory insufficiency data was performed in 20 epochs and repeated 10 times so the results are averaged. We show the performance of each for both MFCC-gram Transformers and Spectrogram Transformers in Table 3.

We observe a small improvement (over the baseline) using time alteration when we test MFCC-gram Transformers including noise in the test files. We also observe an improvement using noise alteration when we test MFCC-gram Transformers without including noise in the test files. In principle, one could combine these techniques as they are independent ways of masking the input. We have done that by performing all three techniques at the same time as shown in the table. Note that the performance of Spectrogram Transformers increases even more robustly than that of MFCC-gram Transformers.

5. Conclusion and Future work

By employing a Transformers network to the dataset of respiratory insufficiency from COVID-19 detection created in the paper [Casanova et al. 2021], we improved the performance of their CNN network from 87.04% to 96.38%. Moreover, we found that MFCC and Spectrogram based Transformers improve their performance through unsupervised pretraining on a large amount of unlabeled data.

Model	Pretraining type	Accuracy (with noise in test samples)	Accuracy (without noise in test samples)
MFCC Transformers	Baseline	96.30 ± 1.12	97.36 ± 1.89
	Time Alteration	$\mathbf{96.53 \pm 0.71}$	97.00 ± 1.55
	Channel Alteration	96.15 ± 0.84	97.04 ± 1.52
	Noise Alteration	95.93 ± 0.66	98.21 ± 0.89
	Time + Channel + Noise	96.38 ± 1.24	$\mathbf{98.54 \pm 1.56}$
Spectrogram Transformers	Baseline	82.84 ± 1.82	82.65 ± 2.25
	Time Alteration	80.99 ± 3.49	87.90 ± 2.75
	Channel Alteration	82.41 ± 1.75	87.53 ± 2.25
	Noise Alteration	80.61 ± 1.32	86.08 ± 2.70
	Time + Channel + Noise	81.67 ± 1.51	86.93 ± 2.61

Table 3. The performance of the Transformers network is compared when unsupervised pretraining is done. The different pretraining techniques are compared for MFCC-gram and Spectrogram Transformers. We fix the amount of noise insertion to be one noise file inserted at patient and control files.

Future work could involve augmenting the dataset with audios from patients of many more respiratory illnesses besides COVID-19. Moreover, we could ideally get audio from patients and control under similar conditions. Furthermore, one could attempt improving the performance of Spectrogram Transformers so that they match the performance of MFCC-gram Transformers. Moreover, we currently train our acoustic model in the single task of respiratory insufficiency detection. It would be interesting to extend our model for other tasks, creating the first acoustic model of spoken Brazilian Portuguese.

6. Acknowledgement

We would like to thank the LNCC for providing us with the computational resources required to do this work. All experiments were run in the LNCC servers. This work was supported by FAPESP grant number 2020/16543-7.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Baevski, A., Schneider, S., and Auli, M. (2019). vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*.

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Botelho, M. C., Trancoso, I., Abad, A., and Paiva, T. (2019). Speech as a biomarker for obstructive sleep apnea detection. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5851–5855. IEEE.

Brigham, E. O. and Morrow, R. (1967). The fast fourier transform. *IEEE spectrum*, 4(12):63–70.

Casanova, E., Gris, L., Camargo, A., Silva, D., Gazzola, M., Sabino, E., Levin, A., Candido Jr, A., Aluisio, S., and Finger, M. (2021). Deep learning against covid-19: Respiratory insufficiency detection in brazilian portuguese speech. *To appear in ACL2021*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gonçalves, S. C. L. (2019). Projeto alip (amostra linguística do interior paulista) e banco de dados iboruna: 10 anos de contribuição com a descrição do português brasileiro. *Estudos Linguísticos (São Paulo. 1978)*, 48(1):276–297.

Gong, Y., Chung, Y.-A., and Glass, J. (2021). Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*.

Laguarta, J., Hueto, F., and Subirana, B. (2020). Covid-19 artificial intelligence diagnosis using only cough recordings. *IEEE Open Journal of Engineering in Medicine and Biology*, 1:275–281.

Liu, A. T., Li, S.-W., and Lee, H.-y. (2020a). Tera: Self-supervised learning of transformer encoder representation for speech. *arXiv preprint arXiv:2007.06028*.

Liu, A. T., Yang, S.-w., Chi, P.-H., Hsu, P.-c., and Lee, H.-y. (2020b). Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6419–6423. IEEE.

Mendes, R. B. (2013). Projeto sp2010: Amostra da fala paulistana. *http://projetosp2010. fflch. usp. br¿. Acesso em*, 1(12):2013.

Nevler, N., Ash, S., Irwin, D. J., Liberman, M., and Grossman, M. (2019). Validated automatic speech biomarkers in primary progressive aphasia. *Annals of Clinical and Translational Neurology*, 6(1):4–14.

Oliviera Jr, M. et al. (2016). Nurc digital um protocolo para a digitalização, anotação, arquivamento e disseminação do material do projeto da norma urbana linguística culta (nurc). *CHIMERA: Revista de Corpus de Lenguas Romances y Estudios Lingüísticos*, 3(2):149–174.

Pham, N.-Q., Nguyen, T.-S., Niehues, J., Müller, M., Stüker, S., and Waibel, A. (2019). Very deep self-attention networks for end-to-end speech recognition. *arXiv preprint arXiv:1904.13377*.

Pinkas, G., Karny, Y., Malachi, A., Barkai, G., Bachar, G., and Aharonson, V. (2020). Sars-cov-2 detection from voice. *IEEE Open Journal of Engineering in Medicine and Biology*, 1:268–274.

Raso, T. and Mello, H. (2012). The c-oral-brasil i: reference corpus for informal spoken brazilian portuguese. In *International Conference on Computational Processing of the Portuguese Language*, pages 362–367. Springer.

Robin, J., Harrison, J. E., Kaufman, L. D., Rudzicz, F., Simpson, W., and Yancheva, M. (2020). Evaluation of speech-based digital biomarkers: Review and recommendations. *Digital Biomarkers*, 4(3):99–108.

Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Song, X., Wang, G., Wu, Z., Huang, Y., Su, D., Yu, D., and Meng, H. (2019). Speech-xlnet: Unsupervised acoustic model pretraining for self-attention networks. *arXiv preprint arXiv:1910.10387*.

Taylor, W. L. (1953). "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Tobin, M. J., Laghi, F., and Jubran, A. (2020). Why covid-19 silent hypoxemia is baffling to physicians. *American journal of respiratory and critical care medicine*, 202(3):356–360.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

DP-Symptom-Identifier: uma estratégia para classificar sintomas de depressão utilizando um conjunto de dados textuais na língua portuguesa

Vinicius Casani[1], Alinne C. Correa Souza[2], Rafael G. Mantovani[3], Francisco Carlos M. Souza[2]

[1]ANYMARKET – Maringá – PR – Brasil
[2]Universidade Tecnológica Federal do Paraná – Dois Vizinhos – PR – Brasil
[3]Universidade Tecnológica Federal do Paraná – Apucarana – PR – Brasil
viniciuscasani3@gmail.com
{alinnesouza, rafaelmantovani, franciscosouza}@utfpr.edu.br

Abstract. *Depression is a psychological disorder that affects millions of people in the world, regardless of their age, social class, or nationality. Different techniques have been studied to analyze and recognize depression in the literature, such as Natural Language Processing, Sentiment Analysis. However, in Brazilian Portuguese, only a few studies have been focused on creating a dataset to classify symptoms of depression. In this paper, we propose a strategy called DP-Symptom-Identifier for collecting twitter messages and generate a novel text dataset with sentences that refer to the main symptoms of depression. Initial experiments with this dataset and different algorithms obtained high accurate performance values, which shows the research in this area is promising.*

Resumo. *A depressão é um distúrbio psicológico que afeta milhões de pessoas no mundo, indiferente à idade, classe social ou nacionalidade. Diferentes técnicas tem sido exploradas para analisar e reconhecer sintomas depressivos na literatura, como técnicas de Processamento de Linguagem Natural e Análise de Sentimentos. Entretanto, para o português brasileiro, poucos estudos tem proposto datasets para a classificação de sintomas da depressão. Neste artigo, propomos uma estratégia chamada DP-Symptom-Identifier para coletar tweets e criar um novo dataset com sentenças que possuem sintomas da depressão. Experimentos iniciais usando diferentes algoritmos obtiveram um alto desempenho preditivo, o que mostra que as pesquisas nesta área são promissoras.*

1. Introdução

Considerada o mal do século pela Organização Mundial da Saúde (OMS) [Organization et al. 2017], a depressão é um transtorno psiquiátrico que afeta o emocional da pessoa. Segundo a OMS, a depressão atinge pessoas de todas as idades, classes sociais e nacionalidades. Além disso, o número de diagnósticos aumentou em $18,4\%$ entre 2005 e 2015, afetando $4,4\%$ da população mundial. No Brasil, a parcela da população afetada é de $5,8\%$, o que coloca o país como o maior detentor do transtorno na América Latina, e o segundo nas Américas, ficando atrás apenas dos Estados Unidos, que tem $5,9\%$ da sua população afetada.

Nesse contexto, técnicas de Análise de Sentimentos (AS) têm sido amplamente utilizadas para detecção e extração de sentimentos a partir de dados textuais. Historicamente a análise de textos é uma tarefa complexa, custosa e tediosa quando realizada manualmente. No entanto, com o atual avanço da tecnologia e o desenvolvimento de áreas como Aprendizado de Máquina (AM) e Processamento de Linguagem Natural (PLN); e uma grande quantidade e disponibilidade de dados gerados por redes sociais, fóruns e páginas na Web, é possível extrair informações riquíssimas e relevantes de dados textuais.

Dentre as vertentes de trabalhos em AS, alguns em específico tem explorado a detecção automática e textual de distúrbios mentais [Hassan et al. 2017, Islam et al. 2018]. Entretanto, esta é ainda uma área de pesquisa emergente, principalmente no Brasil, onde apenas um trabalho relacionado pode ser encontrado [Rosa et al. 2019]. Tendo em vista essa lacuna na literatura, este trabalho tem como objetivo apresentar uma estratégia denominada de *DP-Symptom-Identifier*, a qual foi criada para coletar e identificar os sintomas depressivos em postagens do *Twitter* utilizando técnicas de Processamento de Linguagem Natural (PLN). Para isso, foi desenvolvida uma aplicação Web para coleta, análise e rotulação, indicando as respectivas categorias de sintomas depressivos, como: comportamental, fisiológico e/ou psíquico. Esse processo de rotulação foi acompanhado por uma psicóloga especialista no domínio.

O restante do artigo está estruturado da seguinte forma. Na Seção 2 são descritos os trabalhos relacionados; na Seção 3 é detalhada a estratégia *DP-Symptom-Identifier*; na Seção 4 são apresentados alguns resultados iniciais usando o conjunto de dados gerado, e por fim, na Seção 5 são apresentadas as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Nos últimos anos, as técnicas de AS [Liu 2012] têm recebido atenção especial devido ao seu forte potencial para extrair sentimentos de dados textuais e, consequentemente, auxiliar no diagnóstico de perfis depressivos. Neste contexto, alguns estudos [Birjali et al. 2017, Ma et al. 2017, Aldarwish and Ahmad 2017, Keumhee Kang et al. 2016] utilizaram o *Twitter* como fonte de dados para a criação de base de dados com postagens em inglês. Os dois primeiros estudos realizaram a rotulação dos dados de forma automática, enquanto os demais de forma manual.

Em [Birjali et al. 2017] os autores optaram por buscar postagens do *Twitter* e as rotularam de forma automática utilizando o método de similaridade semântica *WordNet*, tendo como base termos associados com a depressão. Já em [Ma et al. 2017] os autores criaram a base de dados baseado em postagens do *Twitter* contendo a palavra "depressão"; postagens de contas de profissionais de saúdes mental, e de *blogs* sobre depressão. No final do processo, a base de dados compilou um total de 54 milhões de postagens.

No que diz respeito a rotulação manual, no estudo de [Aldarwish and Ahmad 2017] os autores criaram um dataset com 2354 postagens do twitter, 2132 do LiveJournal e 2287 do Facebook. Deste total, 2073 postagens foram rotuladas como deprimidas e 4700 como não deprimidas. Além disso, as postagens identificadas como deprimidas foram sub-categorizadas entre os nove sintomas definidos pelo DSM-5[1]. Por fim, no trabalho de [Keumhee Kang et al. 2016] foram coletadas

[1]Diagnostic and statistical manual of mental disorders

postagens do *Twitter* que continham as seguintes palavras ou sentenças: "*Christmas*", "*Suicide*", "*I feel relaxed*", "*I feel good*", "*want to die*", "*I feel stressed*", "*I feel sad*", "*kill myself*" e "*want to commit suicide*"). Para cada postagem era possível aplicar os seguintes rótulos: negativo, neutro ou positivo.

É importante destacar que apesar do número considerável de estudos, somente o estudo de [Rosa et al. 2019] criou uma base de dados para a língua portuguesa, como também destacado no mapeamento sistemático anteriormente realizado [Casani et al. 2021]. Neste estudo em específico foram filtradas sentenças de postagens do *Facebook* que continham expressões como "odeio minha vida", "me sentindo triste", "estou estressado", entre outras. O estudo também utilizou um conjunto de sentenças positivas para classificar como "não depressão".Entretanto, considerando todos esses estudos, é possível identificar um mesmo problema na criação das bases de dados. A maioria dos estudos concentra-se na identificação de sentimentos positivos e negativos, porém a resolução seguindo uma classificação binária não é suficiente para identificar os sintomas depressivos. Desta forma, em comparação aos trabalhos descritos, o principal diferencial neste artigo é a exploração de uma análise considerando três categorias de textos depressivos na língua portuguesa: sintomas psíquicos, comportamentais e fisiológicos. Segundo o DSM-5 [APA 2013], os sintomas psíquicos e comportamentais são mais relevantes para indicar um quadro depressivo do que os fisiológicos.

3. Estratégia *DP-Symptom-Identifier*

A estratégia *DP-Symptom-Identifier* foi criada com o objetivo de coletar mensagens compartilhadas no *Twitter* utilizando técnicas de PLN. A visão geral da estratégia é apresentada na Figura 1, a qual é composta por cinco etapas: (1) Coleta e rotulação dos dados; (2) Pré-processamento; (3) Extração das características.

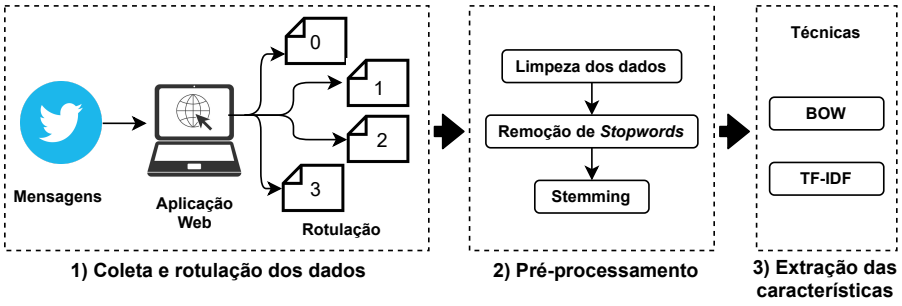


Figura 1. Estratégia *DP-Symptom-Identifier*.

A etapa 1 foi dividida em duas sub-etapas: *(1.1)* coleta dos dados; e *(1.2)* rotulação dos dados coletados. Na primeira sub-etapa (1.1) foi utilizada a API do *Twitter* para coletar as mensagens; e a segunda (1.2) consiste na na rotulação dos dados, por meio de uma aplicação web, classificada em quatro classes possíveis: 0 - Nenhum, 1 - Fisiológico, 2 - Comportamental e 3 - Psíquico. A etapa 2 consiste no pré-processamento dos dados coletados, a fim de reduzir o tamanho da base de dados, bem como remover dados redundantes e irrelevantes utilizando as técnicas de *stopwords* e *stemming*. A etapa 3 visa extrair as características dos dados textuais da base utilizando as técnicas *Bag-of-Words (BoW)* e *Term Frequency-Inverse Document Frequency (TF-IDF)*.

3.1. Coleta e rotulação dos dados

Para realizar a coleta inicialmente foi criada uma base de dados com o auxílio de uma psicóloga para treinar o modelo. Essa base contém 200 sentenças[2] que remetem a sinais sintomáticos da depressão, rotuladas em três categorias: (*i*) comportamental, (*ii*) psíquico e (*iii*) fisiológico. Alguns exemplos de sentenças e seus respectivos rótulos são apresentadas na Tabela 1. É importante salientar que o rótulo dos exemplos pode apresentar quaisquer uma das categorias sintomáticas: Fisiológico, Comportamental, Psíquico, ou ausência completa de sintomas (Nenhum).

Tabela 1. Sintomas presentes em sentenças rotuladas com ajuda da psicóloga.

Sentença	Categoria Sintomática
Eu quero morrer	Comportamental
Não desejo sair de casa	Comportamental
Tenho dificuldade para dormir	Fisiológico
Estou sempre cansado	Fisiológico
Me sinto inútil	Psíquico
Sou infeliz	Psíquico

Para auxiliar a coleta das mensagens foi desenvolvida uma ferramenta em Java, denominada Aplicação de Coleta de Dados (ACD), que utiliza a API do Twitter para buscar mensagens compartilhadas na rede social de acordo com palavras-chave. As funcionalidades implementadas na ACD possibilitaram utilizar palavras e sentenças como parâmetros de filtragem durante a seleção dos *tweets*. Neste contexto, somente postagens que continham os parâmetros desejados eram retornados, como por exemplos as sentenças apresentadas na Tabela 1. Dentre os *tweets* retornados, foram excluídos os que representavam um *retweet*[3].

Uma vez os *tweets* coletados e salvos em disco, os mesmos foram rotulados. Para isso, foi desenvolvida uma segunda Aplicação Web (AW) para que a psicóloga e especialista no domínio definisse os rótulos das postagens de acordo com as categorias sintomáticas apresentadas na Tabela 1. Na Figura 2 é ilustrada a tela referente a análise das rotulações realizadas. Nesta tela, as postagens apresentadas são limitadas a um mesmo usuário, o qual pode ser filtrado pelo campo de texto disponível. Os registros apresentados são ordenados por data da publicação de forma decrescente, simulando uma *timeline* da atividade do usuário ao longo do tempo para facilitar a análise.

A Figura 3 exibe a tela para facilitar o processo de rotulação dos dados coletados. Por meio desta tela é possível rotular cinco postagens por vez, marcando ou desmarcando a opção para cada um dos sintomas. Após a rotulação de todas as postagens, o botão presente no canto inferior direito é acionado para gravar as informações alteradas.

Após o processo de rotulação, cada postagem pode apresentar os seguintes valores para rótulo: *(i)* **Nenhum:** quando a postagem não possui nenhum dos sintomas;

[2]Essa base pode ser acessada no link: https://github.com/fcarlosmonteiro/dp-symptom-identifier

[3]*Retweet* é uma republicação de um *Tweet*.

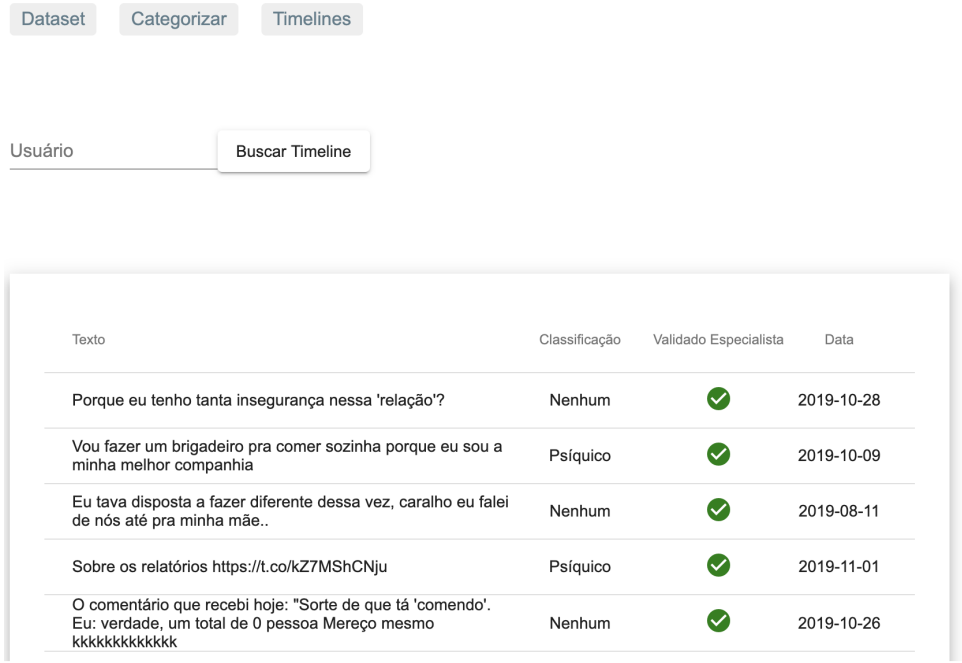


Figura 2. Tela da aplicação AW referente a análise dos dados.

(ii) **Psíquico:** quando a postagem possui apenas indícios de sintomas psíquicos; *(iii)* **Comportamental:** nos casos em que a postagem possui indicação de evidências comportamentais; e *(iv)* **Fisiológico:** quando o sintoma identificado na postagem é de origem fisiológica. Esses valores foram definidos juntamente com a psicóloga e com base nos principais sintomas da depressão. Para a avaliação apropriada foram criadas duas bases de dados distintas: *(i)* Base de treinamento; e *(ii)* Base de dados de testes. A base de dados de treinamento foi criada utilizando as postagens de 1988 usuários coletadas e rotuladas conforme o processo descrito, as quais deveriam estar em conformidades com o padrão das sentenças apresentadas na Tabela 1. Esta base possui um total de 2008 linhas e 7 coluna[4]. A Figura 4 demonstra um exemplo desta base.

Além do dado textual extraído da postagem "tweet", as demais colunas descritivas são: "idTweet" coluna utilizada para identificação dos *tweets*, "fisiologico" coluna a qual remete se o *tweet* pertence a categoria de sintomas fisiológicos. A coluna "comportamental" é utilizada para controlar o fato do *tweet* ser ou não um sintoma da categoria comportamental. Já a coluna "psiquico" demonstra a presença ou não de sintomas psíquicos. É importante ressaltar que o mesmo *tweet* pode ser classificado em mais de uma categoria. A coluna "label" contém informações referente a rotulação do dado. A coluna "dhTweet" apresenta e data da postagem. Por fim a coluna "validadoEspec" informa se a postagens foi validade pela especialista que acompanha este trabalho.

Já a base de dados de testes foi criada utilizando a funcionalidade de busca de *timelines* por ID de usuário, existente na ACD. Esta base possui as mesmas colunas que

[4]Essa base pode ser acessada no link:https://github.com/fcarlosmonteiro/dp-symptom-identifier

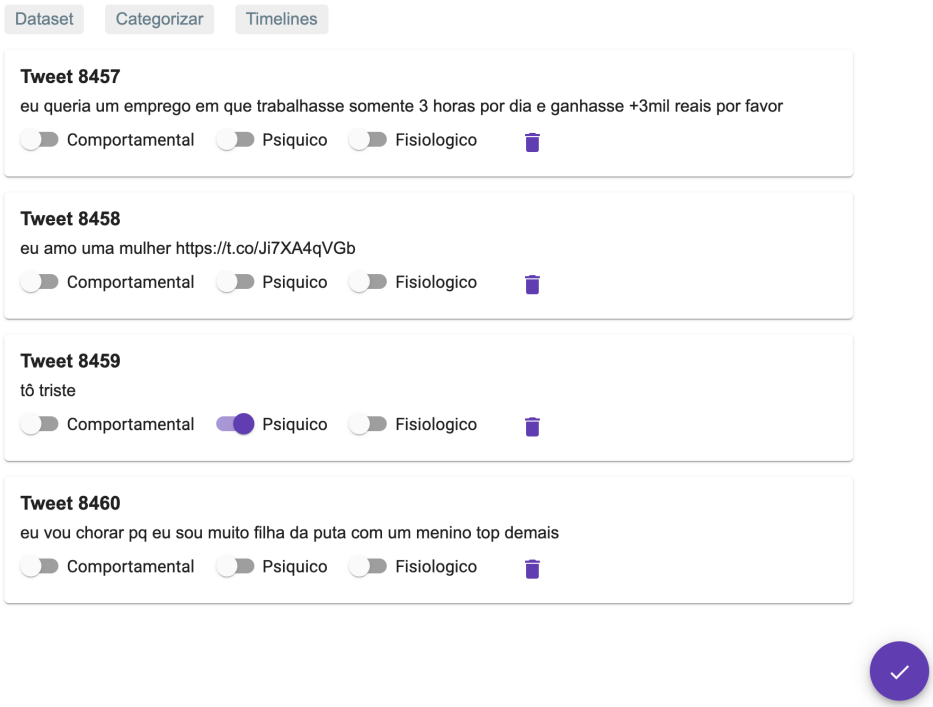


Figura 3. Tela da aplicação AW referente a rotulação dos dados.

	tweet	idTweet	fisiologico	comportamental	psiquico	label	dhTweet	validadoEspec
1	só eu que já fiquei com dúvida se só tô muito triste o...	11	FALSE	FALSE	TRUE	3	2019-09-12	TRUE
2	Crises de ansiedade Depressão Tô me afundando e n ...	20	FALSE	FALSE	FALSE	0	2019-09-11	TRUE
3	hj vai passar uma materia sobre depressao no globo r...	22	FALSE	FALSE	FALSE	0	2019-09-11	TRUE
4	Hoje o Profissão Repórter fala sobre Depressão. ASSI...	23	FALSE	FALSE	FALSE	0	2019-09-11	TRUE

Figura 4. Exemplo da base de dados de Treinamento.

a base de dados de treinamento. Os dados armazenados nesta base podem ser utilizados para testar modelos preditivos quando fazendo o uso de algoritmos de AM, para classificar os dados. Neste contexto, a quantidade de subconjuntos (k) escolhida para validar os modelos criados é de $k = 10$ subconjuntos, valor frequentemente utilizado pela literatura.

Por fim, a Figura 5 demonstra a distribuição das classes existente na base de dados de treinamento. A classe que representa os sintomas psíquicos é a que possui a maior quantidade de exemplos dentre as demais. Isso pode ser justificado pelo fato de que esta categoria sintomática é a que possui uma maior quantidade de exemplos no *American Psychiatric Association* [APA 2013], e ainda, alguns sintomas da categoria comportamental são relacionados com os da categoria psíquico.

3.2. Pré-processamento

Esta etapa é realizada com a finalidade de reduzir o tamanho da base de dados, e também remover dados redundantes e irrelevantes. Em uma primeira etapa todos os caracteres das postagens foram convertidos para caracteres minúsculos. Em seguida, caracteres

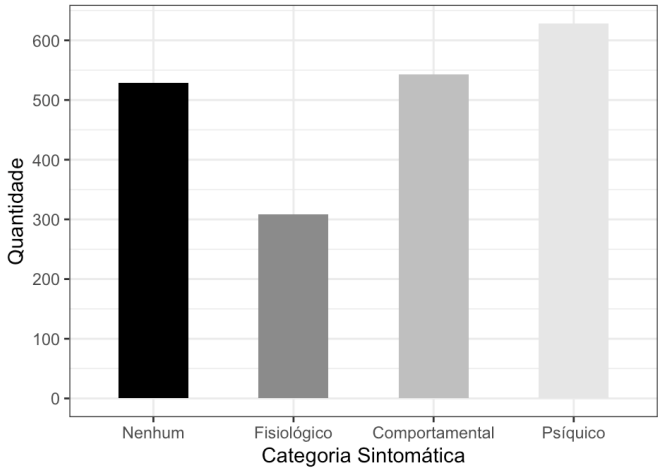


Figura 5. Distribuição das classes presentes no conjunto de treinamento.

numéricos foram removidos, assim como pontuações, *links* e espaços desnecessários. Além disso, citações de outros usuários e *retweets* também foram removidos. Um último passo removeu todos os caracteres especiais, tais como *emojis*, pois estes não representam evidências sintomáticas da depressão. Em seguida, duas técnicas de PLN foram utilizadas: remoção de *stopwords* - palavras que devem ser desconsideradas na análise pois não apresentam informações relevantes para a construção do modelo que será analisado, como por exemplo "o", "e", "a", "de", "que"; e *stemming* para a redução das palavras ou termos ao seu radical.

3.3. Extração de Características

Para identificar a melhor forma de extrair as características dos dados textuais da base, foram realizados testes com duas técnicas: *(i) (TF-IDF)* e *(ii) BoW*. A técnica TF-IDF visa expressar a relevância de uma palavra em um dado *corpus*, a qual foi utilizada com o auxílio da função `TfIdfVectorizer` do pacote `superml`[5] do R. Ao executar a vetorização dos 2008 exemplos presentes na base da dados, o processo retornava uma matriz com mais de 10 mil atributos descritivos (colunas). Para tentar diminuir esta quantidade, foram identificados e removidos todos os atributos correlacionados, em uma porcentagem igual ou superior a 95%. Mesmo após realizar esta operação de redução, ainda ocorria a geração de um *dataset* longitudinal, no qual a quantidade de atributos (colunas) é maior que a quantidade de exemplos (linhas) disponível na base de dados. Nos primeiros testes realizados, tais bases longitudinais geravam modelos preditivos com *overfiting*: eles prediziam sempre a mesma classe (majoritária) e erravam a predição de todas as outras classes.

Por conta desses resultados uma segunda técnica foi utilizada, o *Bag-of-Words* (BoW), que consiste em a frequência que cada termo se repete dentro da base de dados. Usando o BoW a quantidade de atributos descritivos gerados foi de 4650. Após o processamento principal da técnica, foram removidos os atributos com uma frequência menor

[5]`https://www.rdocumentation.org/packages/superml/versions/0.4.0`

que 10, reduzindo a quantidade total de atributos descritivos para 340. Tal mudança foi muito significativa, pois foi possível induzir modelos mais precisos, obtendo resultados superiores em até 50% quando comparado com os resultados obtidos com *TF-IDF*. Sendo assim, o *BoW* foi a técnica newadotada para realizar a extração de características dos dados textuais.

4. Experimentos iniciais para avaliação do conjunto de dados

Com a base criada, um experimento pequeno foi realizado no intuito de avaliar a qualidade do conjunto de dados. O *setup* experimental definido utilizou apenas o conjunto de treinamento descrito nas seções anteriores, e quatro diferentes algoritmos de classificação: *Support Vector Machines* (SVMs), *Näive Bayes* (NB), *Multilayer Perceptron (MLP)* e *Random Forest* (RF). Os algoritmos foram executados por meio de uma validação cruzada usando 10 partições de dados estratificadamente amostrados. Foram criadas múltiplas tarefas preditivas binárias para cada categoria sintomática. O desempenho dos modelos induzidos foi mensurado por meio da AUC[6], e realizada uma média para cada um dos algoritmos. Os resultados gerais são apresentados na Tabela 2.

Tabela 2. Valores de AUC obtidos pelos algoritmos avaliados no conjunto de treinamento gerado pela estratégia

Algoritmo	AUC	sd
RF	0.935	0.015
SVM	0.927	0.013
MLP	0.900	0.009
NB	0.725	0.020

Os resultados iniciais obtidos sugerem que a estratégia elaborada é descritiva e efetiva, pois as características textuais extraídas são mapeadas corretamente para as categorias sintomáticas da depressão. Isso é visivelmente corroborado pelos valores médios de AUC acima de 0.9, obtidos pelos algoritmos RF, SVM e MLP.

5. Considerações finais

Neste trabalho foi desenvolvida a estratégia *DP-Symptom-Identifier* para coletar e analisar mensagens compartilhadas em português no *Twitter* a fim de identificar um dos três tipos de sintomas da depressão (Psíquicos, Fisiológicos e Comportamentais). A estratégia é composta por uma aplicação web que facilita o processo de coleta das postagens do *Twitter* e a rotulação dos dados, a qual foi realizada com o auxílio de uma psicóloga. Além disso, foi possível analisar as postagens com um experimento inicial, e comprovar a efetividade da proposta. Como trabalhos futuros pretense-se: expandir o conjunto de dados, adicionando mais *tweets*; rotula-los usando as ferramentas desenvolvidas; e incrementar os experimentos com os algoritmos preditivos. Deseja-se também expandir a coleta de dados para outras redes sociais.

[6]É uma medida de desempenho utilizada em problemas de classificação, que representa a medida de separabilidade, ou seja, mostra até que ponto o modelo avaliado é capaz de separar corretamente as classes [Flach et al. 2011].

Referências

Aldarwish, M. M. and Ahmad, H. F. (2017). Predicting depression levels using social media posts. In *2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS)*, pages 277–280.

APA (2013). *Diagnostic and statistical manual of mental disorders : DSM-5*. American Psychiatric Association Arlington, VA, 5th ed. edition.

Birjali, M., Beni-Hssane, A., and Erritali, M. (2017). A method proposed for estimating depressed feeling tendencies of social media users utilizing their data. pages 413–420. Springer International Publishing.

Casani, V., Mantovani, R. G., Souza, A. C. C., and Souza, F. C. M. (2021). Identificação de perfis depressivos em redes sociais utilizando aprendizado de máquina: um mapeamento sistemático. *Computer on The Beach*, (12).

Flach, P., Hernandez-Orallo, J., and Ferri, C. (2011). A coherent interpretation of auc as a measure of aggregated classification performance. pages 657–664.

Hassan, A. U., Hussain, J., Hussain, M., Sadiq, M., and Lee, S. (2017). Sentiment analysis of social networking sites (sns) data using machine learning approach for the measurement of depression. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 138–140.

Islam, M. R., Kamal, A. R. M., Sultana, N., Islam, R., Moni, M. A., and ulhaq, A. (2018). Detecting depression using k-nearest neighbors (knn) classification technique. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, pages 1–4.

Keumhee Kang, Chanhee Yoon, and Eun Yi Kim (2016). Identifying depressive users in twitter using multimodal analysis. In *2016 International Conference on Big Data and Smart Computing (BigComp)*, pages 231–238.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Ma, L., Wang, Z., and Zhang, Y. (2017). Extracting depression symptoms from social networks and web blogs via text mining. pages 325–330. Springer International Publishing.

Organization, W. H. et al. (2017). Depression and other common mental disorders: global health estimates. Technical report, World Health Organization.

Rosa, R. L., Schwartz, G. M., Ruggiero, W. V., and Rodríguez, D. Z. (2019). A knowledge-based recommendation system that includes sentiment analysis and deep learning. *IEEE Transactions on Industrial Informatics*, 15(4):2124–2135.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Identificando sintomas de depressão em postagens do Twitter em português do Brasil

Augusto R. Mendes, Rafael V. P. Passador, Helena M. Caseli

[1]Universidade Federal de São Carlos (UFSCar)
Departamento de Computação – LALIC
Caixa Postal 676 – 13565-905 – São Carlos – SP – Brasil

augustorm@estudante.ufscar.br,rafaelpassador@estudante.ufscar.br

AVISO: Este documento contém conteúdo sensível, com exemplos de termos e descrição de sintomas de transtornos depressivos.

Abstract. *Currently, depression is one of the most worrisome mental health issues. In Brazil, in 2019, 10.2% of the adult population reported having been diagnosed with depression according to data from the National Health Survey. Identifying people with a possible depressive profile allows adequate monitoring by mental health professionals. In this sense, online social networks such as Twitter can be important allies. This article presents experiments carried out for the automatic classification of Twitter posts (not users) containing content that denotes some symptom of depression. Logistic regression showed the best results (average* $F1$ *equal to* 57%*) among the investigated algorithms.*

Resumo. *A depressão é uma das questões de saúde mental mais preocupantes da atualidade. No Brasil, em 2019, 10,2% da população adulta relatou ter sido diagnosticada com depressão segundo dados da Pesquisa Nacional de Saúde. Identificar pessoas com perfil possivelmente depressivo permite um acompanhamento adequado por parte dos profissionais de saúde mental. Nesse sentido, as redes sociais online, como o Twitter, podem ser importantes aliadas. Este artigo apresenta experimentos realizados para a classificação automática de postagens (e não usuários) do Twitter contendo conteúdo que denota algum sintoma de depressão. A classificação com regressão logística apresentou os melhores resultados (*$F1$ *média de* 57%*) entre os algoritmos investigados.*

1. Introdução

O tratamento de depressão pode ser considerado uma das mais importantes questões de saúde mental da atualidade. No Brasil, segundo dados da Pesquisa Nacional de Saúde realizada em 2019, 10,2% dos adultos brasileiros mencionaram terem recebido diagnóstico de depressão por profissional de saúde mental[1].

A depressão maior pode ser diagnosticada quando uma pessoa apresenta, por pelo menos duas semanas, cinco ou mais sintomas, sendo que pelo menos um dos sintomas é humor deprimido ou perda de interesse ou prazer; os outros sintomas compreendem: diminuição ou aumento de peso ou apetite; insônia ou hipersonia; agitação ou retardo psicomotor; fadiga ou perda de energia; sentimentos de inutilidade ou culpa; concentração

[1]Disponível em: https://www.ibge.gov.br/estatisticas/sociais/saude/9160-pesquisa-nacional-de-saude.html?edicao=29270&t=resultados

diminuída, ou indecisão e/ou pensamentos recorrentes de morte. Segundo o DSM-5 [American Psychiatric Association et al. 2014], esses sintomas devem causar sofrimento clinicamente significativo ou prejuízo no funcionamento social, profissional ou em outras áreas importantes da vida do indivíduo.

O rastreio de sintomas depressivos e de funcionalidade, podem ser realizados por instrumentos auto aplicáveis, ajudando a fornecer informações sobre o humor e a funcionalidade do indivíduo em um recorte de tempo. Para o rastreio de sintomas depressivos são usados mecanismos como a escala *Beck Depression Inventory* II (BDI-II) [BECK et al. 1961] e para a funcionalidade, a Escala de Avaliação de Incapacidade da Organização Mundial da Saúde 2.0 (WHODAS 2.0)[2].

Identificar pessoas com Perfil Possivelmente Depressivo (PPD) possibilita intervenções e acompanhamento por profissionais da área da saúde e é essencial para o direcionamento efetivo de recursos de saúde mental. Nesse sentido, as Redes Sociais Online (RSO) podem ser importantes aliadas. Estudos como os de [De Choudhury et al. 2013] apontam que RSO são ambientes mais naturais para identificação de PPD quando comparados com os instrumentos comumente usados na área de saúde.

Na língua inglesa, existem diversos exemplos bem-sucedidos de técnicas de Processamento de Linguagem Natural (PLN) aplicadas para saúde mental, como a previsão de diagnósticos clínicos de depressão de usuários do Facebook através de suas postagens pré-diagnóstico [Eichstaedt et al. 2018], a detecção de ideação suicida em postagens no Reddit [Ji et al. 2018] e Twitter [O'Dea et al. 2015] e a classificação de notas de suicídio genuínas e falsas [Pestian et al. 2012].

No caso da língua portuguesa, a literatura é consideravelmente mais escassa. Em [Duque et al. 2018], os autores relatam a classificação de sentenças depressivas no Twitter. Em [Santos et al. 2020], descreve-se a construção de um córpus que visa apoiar tanto o reconhecimento das questões de saúde mental em mídias sociais *online*, quanto a análise temporal dessas doenças, com a apresentação de resultados iniciais de classificação de textos abordando ambas as tarefas.

Nesse contexto, este artigo apresenta experimentos realizados para a classificação automática de postagens (e não usuários) com conteúdo que denota algum sintoma de depressão. Para tanto, o córpus de [Santos et al. 2020], que não está anotado com sintomas, foi anotado via um processo de rotulação fraca com base em uma versão traduzida do léxico de [Yazdavar et al. 2017].

As principais contribuições deste trabalho são: (i) primeiro trabalho de classificação automática de sintomas de depressão no português do Brasil, (ii) versão em português do léxico de [Yazdavar et al. 2017], (iii) versão do córpus de [Santos et al. 2020] anotada (via rotulação fraca) com sintomas de depressão, e (iv) classificador automático de sintomas de depressão com $F1$ média de 57%.

2. Trabalhos relacionados

Na literatura foram propostos diferentes métodos em relação ao escopo, plataformas analisadas, padrão-ouro (*gold standard*), engenharia de *features* e modelos usados. Em geral,

[2]Disponível em: `https://apps.who.int/iris/bitstream/handle/10665/43974/9788562599514_por.pdf?sequence=19`

os trabalhos comprovam que é possível captar informações relevantes sobre a saúde mental dos usuários a partir do conteúdo textual publicado por eles nas mídias sociais.

Em [Mowery et al. 2016], os autores investigaram a prevalência dos sintomas associados ao transtorno depressivo maior ao longo do tempo, nos Estados Unidos, com base em dados do Twitter. Para tanto, desenvolveram classificadores automáticos para discernir se um *tweet* apresenta (ou não) evidência de depressão com base em diversas *features* binárias que indicam a presença ou ausência de: (1) n-gramas que representam termos relacionados a um sintoma, (2) etiquetas de *part-of-speech*, (3) emoticons que podem expressar visualmente algum sintoma depressivo, (4) idade/gênero do autor do *tweet*, características inferidas por meio da frequência de termos agregados em um léxico, (5) grau de subjetividade e polaridade da postagem, (6) traços de personalidade e (7) outras características extraídas do LIWC. O SVM foi o algoritmo que apresentou o melhor desempenho geral para a classificação automática com base em $F1$, e obteve performance variável na detecção de sintomas individuais indo de 35% para a detecção do sintoma "humor deprimido" até 75% no sintoma "fadiga ou perda de energia".

Em [Ziwei e Chua 2019], utilizou-se uma aplicação web para classificação de postagens depressivas publicadas no Twitter, em inglês, por meio da análise de palavras potencialmente depressivas, neutras e não depressivas presentes nos léxicos de [Yazdavar et al. 2017]. Esse mesmo recurso (léxicos) é utilizado em outros trabalhos, como [Yadav et al. 2020], onde os autores propuseram uma abordagem para identificar sintomas depressivos utilizando BERT (*Bidirectional Representation from Transformers*) e aprendizado multitarefa (*multitask learning*). Para tanto, os autores coletaram e anotaram um córpus composto por 12.155 *tweets* de usuários depressivos, com 3.738 deles anotados com as 9 classes de sintomas da escala PHQ-9 (*Patient Health Questionnaire*-9) obtendo uma alta precisão ($96,91\%$) na detecção de postagens depressivas. A identificação dos sintomas, pelo método proposto, obteve um $F1$ médio de 75% superando em pelo menos 2 pontos percentuais os métodos usados na comparação.

Em [Santos et al. 2020], os autores reiteram a busca por sinais de problemáticas relacionadas à saúde mental no Brasil. Para tanto, utilizaram um córpus composto por postagens de usuários do Twitter (detalhado na seção 3.2) que foram, então, divididas em: (i) postagens escritas antes do diagnóstico/tratamento e (ii) postagens escritas durante/depois do evento. Os autores utilizaram recursos linguístico-computacionais (LIWC, *word embeddings*), TF-IDF e algoritmos de AM (regressão logística, MLP). Os classificadores comparados foram: regressão logística com TF-IDF, *multilayer perceptron* com média de *word embeddings* e regressão logística com frequências de categorias do LIWC. Conclui-se que regressão logística com TF-IDF supera as alternativas (com $F1$ médio de 69% contra 59% alcançado pelo segundo melhor método, MLP com *word embeddings*).

Este artigo apresenta experimentos desenvolvidos com o córpus de [Santos et al. 2020] e uma versão traduzida do léxico de [Yazdavar et al. 2017] para a identificação automática dos 9 sintomas de depressão da PHQ-9. Todos esses recursos usados neste trabalho são apresentados na seção 3.

3. Recursos

Esta seção descreve os recursos utilizados neste trabalho para identificar automaticamente os 9 sintomas de depressão da PHQ-9[3] [Spitzer et al. 1999]: (1) falta de interesse, (2) tristeza/humor depressivo, (3) desordem de sono, (4) falta de energia, (5) desordem alimentar, (6) baixa auto-estima, (7) problemas de concentração, (8) hiperatividade/baixa atividade e (9) pensamentos de suicídio.

3.1. Lista semente de [Yazdavar et al. 2017] traduzida para o português

Com a ajuda de um profissional de psicologia clínica, em [Yazdavar et al. 2017] foi produzida uma lista de termos relacionados a cada sintoma de depressão avaliado pelo PHQ-9. Para tanto, os construtores da lista usaram a ferramenta Big Huge Thesaurus[4] e o dicionário colaborativo de gírias Urban Dictionary[5] para ajudar na busca por sinônimos. O léxico final contém mais de 1.620 entradas em inglês relacionadas à depressão. Além dos 9 sintomas da PHQ-9, os autores consideraram um décimo em seu léxico: (10) medicamentos relativos a depressão.

Como parte do presente trabalho, a lista original (em inglês) foi traduzida para o português por dois nativos do português, com bons conhecimentos em inglês e com auxílio do tradutor automático Google Translate[6]. O processo consistiu na entrada dos termos e subsequente avaliação dos resultados, levando em consideração o sintoma associado, bem como adequação com o vocabulário cotidiano. Nos casos em que o tradutor não forneceu saída adequada a partir da entrada fora de contexto, buscou-se traduzir uma sentença completa na qual a entrada era usada em um contexto relevante ao sintoma[7].

Ao final deste processo, um conjunto de 1.213 *seeds*, em português, foi obtido. Diferenças no número de *seeds* em relação ao conjunto original têm duas causas. Primeiro, o conjunto continha expressões coloquiais (como "blothpick"), que nem sempre têm traduções óbvias para o português, foram descartadas. Outro fator para a diferença de quantidade é que várias entradas diferentes no inglês acabavam tendo a mesma tradução. Além disso, boa parte dos termos do léxico original têm gênero neutro (como "*depressed*"), resultando na geração de mais termos ao traduzir ("deprimido" e "deprimida", por exemplo).

A Tabela 1 traz exemplos de palavras e expressões associados a cada um dos sintomas, que no léxico original e no traduzido são identificados como "sinais" (*signals*).

3.2. Córpus de [Santos et al. 2020]

Em [Santos et al. 2020], os autores coletaram um córpus do Twitter que consiste em usuários que relataram terem sidos diagnosticados com transtornos de saúde mental por profissionais da saúde, ou que relataram terem iniciado os tratamentos para uma dessas

[3]O *Patient Health Questionnaire* (PHQ-9) (`https://images.app.goo.gl/G6VoWfHL2Y4yVvLaA`) é um questionário de 9 itens usado para diagnóstico de depressão, voltado para rápida aplicação por clínicos gerais.

[4]Disponível em: `https://words.bighugelabs.com/`

[5]Disponível em: `https://www.urbandictionary.com/`

[6]Disponível em: `https://translate.google.com/`

[7]Por exemplo, o tradutor automático gerou a equivalente "blues" para a palavra "bluesy", mas quando a mesma palavra foi empregada na sentença "I'm feeling bluesy" o tradutor foi capaz de gerar a tradução "Estou me sentindo triste".

Tabela 1. Exemplos de entradas no léxico de sintomas traduzido para o português

Sinal	Sintoma associado	Exemplo	n° de termos
1	falta de interesse	sem_graça, apatia, tédio	178
2	tristeza/humor depressivo	melancólico, deprimente, abatido	192
3	desordem de sono	insônia, sonolento, sem_dormir	82
4	falta de energia	não_tenho_força, preguiça, cansado	95
5	desordem alimentar	odeio_minhas_coxas, anorexia, barrigudo	146
6	baixa auto-estima	não_mereço, desprezível, eu_me_odeio	112
7	problemas de concentração	dispersa, distraído, desorientado	96
8	hiperatividade/baixa atividade	ansiedade, agitado, inquieto	81
9	pensamentos de suicídio	me_matar, merece_morrer, sono_eterno	104
10	medicamentos	lítio, alprazolam, citalopram	127

condições. Para tanto, a coleta se baseou em termos relacionados à saúde mental (como depressão e ansiedade) e ao diagnóstico, tratamento, ou o uso de medicamentos antidepressivos.

Em seguida, todas as mensagens coletadas foram manualmente inspecionadas para filtrar aquelas que pareciam suficientemente genuínas. As 3.200 mensagens mais recentes dos autores foram, então, rotuladas apenas no nível do usuário. Para cada autor selecionado, as mensagens foram examinadas de modo a identificar o momento específico em que o diagnóstico ou tratamento foi iniciado, e para destacar o subconjunto de mensagens que foram publicados antes desse evento.

Criou-se também um grupo de controle constituído por usuários que manifestaram interesse em questões de saúde mental como (i) uma preocupação geral (por exemplo, promovendo a campanha de prevenção de suicídio "Setembro Amarelo"), (ii) uma preocupação em relação a uma pessoa particular que sofria de um problema de saúde mental (por exemplo, um amigo), ou (iii) por ser um estudante de psicologia com um interesse no tema da depressão. As mensagens obtidas em resposta a essas consultas foram inspecionadas manualmente para remover usuários diagnosticados ou tratados para problemas de saúde mental.

4. Experimentos

Essa seção descreve as etapas do experimento realizado para classificar automaticamente os 9 sintomas da PHQ-9 no córpus de [Santos et al. 2020].

4.1. Rotulação fraca do córpus

Com o intuito de realizar a identificação automática de sintomas de depressão adotou-se a estratégia de rotulação fraca.[8] Tal estratégia foi adotada porque não há disponível livremente um córpus anotado com tais sintomas para o português do Brasil, como havia no caso de [Yadav et al. 2020].

Assim, a partir do córpus de [Santos et al. 2020], no qual cada postagem está associada a uma de duas classes (positiva ou negativa) para o perfil depressivo, uma rotulação

[8]A rotulação fraca é uma técnica para rotulação automática de um córpus a partir de casamento de padrão de uma lista de termos com os textos a serem rotulados

fraca foi aplicada para anotar os sintomas. Neste processo de rotulação fraca, os novos rótulos de uma dada postagem são atribuídos com base em dois fatores: (i) a classe atribuída para a tarefa de detecção de depressão, e (ii) a presença de palavras-chave associadas a um dado sintoma. Assim, apenas as postagens rotuladas como positivas (PPD) no córpus de [Santos et al. 2020] foram processadas para associar, via rotulação fraca, a(s) classe(s) do(s) sintoma(s) que contém. A Tabela 2 traz as quantidades de instâncias para cada sintoma.

Tabela 2. Quantidade de instâncias para cada sintoma

Sintoma	# instâncias
1 - falta de interesse	5.052
2 - tristeza/humor depressivo	17.218
3 - desordem de sono	1.400
4 - falta de energia	4.854
5 - desordem alimentar	1.616
6 - baixa auto-estima	7.380
7 - problemas de concentração	2.162
8 - hiperatividade/baixa atividade	4.146
9 - pensamentos de suicídio	10.190

4.2. Pré-processamento do córpus

Tratando-se de textos produzidos para mídias sociais, as postagens apresentam muitas abreviações, erros ortográficos, links e artefatos típicos de uma dada plataforma (por exemplo, menções no formato "@username" no Twitter). Estas características constituem ruído e afetam negativamente a extração de *features* e posterior treinamento de modelos. Portanto, foi usada a ferramenta Enelvo [Bertaglia e Nunes 2016] para a normalização de abreviações (por exemplo, "vc" é normalizado para "você"), erros ortográficos e identificação de ruídos (por exemplo, urls são todas normalizadas para a forma "url"), que foram subsequentemente removidos.

4.3. Extração de *features*

As seguintes *features* foram utilizadas pelos algoritmos de aprendizado de máquina:

LIWC – Utilizando o LIWC para o português[9], foi feita a contagem do número de termos, em cada postagem, que pertenciam às seguintes categorias: *family, anx, sad, ingest, work, money, death, friend, health*. Dessa forma, visou-se capturar informação a respeito de fatores sociais, econômicos e outros relacionados a sintomas como alimentação. Ao final deste processo, nove *features* binárias indicando presença (1) ou ausência (0) de termos em cada categoria foram geradas.

TF-IDF – Foi feita a extração da frequência de cada termo, em cada postagem, com contra-peso de acordo com a frequência de postagens como um todo (TF-IDF), com o intuito de capturar a temática e características de uma dada postagem que a diferencia das demais. Um vetor numérico contendo os valores de TF-IDF foi gerado neste processo.

[9]Disponível em: `http://143.107.183.175:21380/portlex/index.php/pt/projetos/liwc`

Polaridade – Neste trabalho, realizou-se o ajuste fino do modelo BERTimbau [Souza et al. 2020] com o córpus TweetsentBR [Brum e Nunes 2018], tendo como resultado um classificador de polaridade que atingiu 67% de precisão no treinamento. Este modelo foi, então, usado para determinar a polaridade de uma postagem como a polaridade majoritária das sentenças que a compõem.[10] Uma *feature* categórica (-1 para negativa, 0 para neutra e 1 para positiva) foi gerada neste processo.

Embeddings – Utilizando as NILC-embeddings[11], modelo CBOW de dimensão 100, foram calculados os vetores para cada palavra em uma dada postagem. A representação da postagem foi gerada como a média dos vetores de palavra. Além das *embeddings* pré-treinadas, foram usadas também *embeddings* treinadas a partir do córpus (por meio da biblioteca `gensim`[12]), com peso determinado pelo valor TF-IDF de cada palavra aplicado ao vetor da mesma. Assim, ao final deste processo, duas representações para a postagem foram usadas como *features*: uma baseada nas NILC-embeddings e outra baseada nas *embeddings* treinadas para o córpus deste trabalho.

As *features* extraídas foram, então, usadas pelos algoritmos SVM, regressão logística e MLP. O treinamento dos modelos foi realizado via `scikit-learn`[13]. Foi feita redução de dimensionalidade das *features* por meio de PCA, seguindo o seguinte critério: dos 50 componentes extraídos, foram incluídos todos aqueles que explicassem cumulativamente 90% de variância, sendo o restante descartados. Foi feita amostragem do córpus de forma que os conjuntos de treino e teste fossem balanceados.

5. Resultados

A Tabela 3 resume as configurações de melhor resultado, em termos de $F1$, para cada sintoma. Esses valores foram obtidos considerando-se a validação cruzada (5-*fold*) no treinamento dos modelos a partir do conjuntos de instâncias (Tabela 2) para cada sintoma, com diferentes combinações de *features*. Estas configurações estão listadas na tabela no formato: Algoritmo usado + combinação de *features* entre parênteses (e.g: LogReg(TF-IDF + LIWC).

A regressão logística (LogReg) foi o algoritmo de melhor desempenho na classificação de 6 das 9 classes de sintomas. As *features* TF-IDF e LIWC se destacaram, enquanto o impacto da polaridade do texto na performance é inconclusivo (como pode-se observar na Tabela 3). A comparação direta com os trabalhos relacionados não é indicada neste caso, uma vez que idioma e modo de anotação (rotulação fraca) adotados aqui diferem dos trabalhos da literatura. Contudo, vale mencionar que este trabalho estende [Santos et al. 2020] e corrobora sua constatação de que regressão logística com TF-IDF tem bom desempenho na análise de postagens com PPD.

É importante ressaltar que a estratégia de rotulação fraca apresenta limitações. Apesar de ela se basear na presença de diagnóstico prévio e na ocorrência de palavras e expressões associadas aos sintomas, não é possível garantir que a postagem em questão

[10]Em caso de empate, adotou-se a polaridade neutra.
[11]Disponível em: `http://www.nilc.icmc.usp.br/embeddings`
[12]Disponível em: `https://radimrehurek.com/gensim/`
[13]Disponível em: `https://scikit-learn.org/`

Tabela 3. Melhores resultados de $F1$ para cada sintoma

Sintoma	Melhor valor de $F1$	Configuração do melhor resultado
1 - falta de interesse	54,66%	LogReg (TF-IDF + LIWC + polaridade)
2 - tristeza/humor depressivo	56,09%	LogReg (TF-IDF + LIWC + polaridade)
3 - desordem de sono	56,47%	SVC (*embeddings* com peso)
4 - falta de energia	58,59%	LogReg (TF-IDF + LIWC)
5 - desordem alimentar	60,22%	LogReg (TF-IDF + LIWC + polaridade)
6 - baixa auto-estima	55,35%	SVC (*embeddings* com peso)
7 - problemas de concentração	55,83%	LogReg (TF-IDF + LIWC + polaridade)
8 - hiperatividade/baixa atividade	58,11%	SVC (*embeddings* com peso)
9 - pensamentos de suicídio	57,66%	LogReg (TF-IDF + LIWC)
Média	56,99%	–

relata mesmo o sintoma. A rotulação fraca é especialmente prejudicada pela presença de ambiguidade e pela subjetividade da tarefa, uma vez que nem toda postagem produzida por uma pessoa deprimida necessariamente evidencia sintoma de depressão.

Nesse ínterim, observa-se a incidência de falsos positivos e falsos negativos. Por exemplo, analisando as anotações feitas para o sintoma "desordem de sono" pelo SVC (*embeddings* com peso), as postagens classificadas que continham os termos "cansada", "preguiça" e "cansaço" apresentaram maiores ocorrências de falsos positivos.

6. Conclusões e Trabalhos futuros

Como discutido na seção 5, a rotulação fraca apresenta limitações especialmente no contexto de tarefas como a investigada neste artigo, na qual o conteúdo deve ser avaliado considerando critérios subjetivos e instrumentos da saúde mental. Assim sendo, um dos trabalhos futuros será realizar a anotação do córpus com o auxílio de especialistas em saúde mental, uma vez que esta parece ser a melhor estratégia dadas as características da tarefa proposta. Também como trabalho futuro, pretende-se submeter a lista semente traduzida para a revisão por especialistas em saúde mental e linguística, de forma a validar o recurso para a língua portuguesa, garantindo maior qualidade de tradução.

Mesmo com as presentes limitações, os experimentos conduzidos produziram modelos capazes de classificar sintomas com $F1$ entre 55% e 60%, indicando a viabilidade da tarefa. Trabalhos futuros buscarão investigar novos conjuntos de *features* e técnicas de classificação, como a utilização de modelos pré-treinados baseados em *transformers* [Yadav et al. 2020].

Todos os recursos e códigos produzidos nesta pesquisa estão disponíveis em `https://github.com/LALIC-UFSCar/Amive-PLN`

Agradecimentos

Agradecemos ao professor Ivandré Paraboni pela disponibilização do córpus utilizado neste trabalho. Agradecemos também o suporte financeiro das entidades de apoio e incentivo à pesquisa acadêmica: o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP). Em especial as bolsas PIBIC e IC FAPESP (#2021/02430-9) e o projeto Amive (Auxílio Regular FAPESP #2020/05157-9) do qual esta pesquisa faz parte.

Referências

American Psychiatric Association et al. (2014). *DSM-5: Manual diagnóstico e estatístico de transtornos mentais*. Artmed Editora.

BECK, A. T., WARD, C. H., MENDELSON, M., MOCK, J., e ERBAUGH, J. (1961). An Inventory for Measuring Depression. *Archives of General Psychiatry*, 4(6):561–571.

Bertaglia, T. F. C. e Nunes, M. d. G. V. (2016). Exploring word embeddings for unsupervised textual user-generated content normalization. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 112–120.

Brum, H. e Nunes, M. d. G. V. (2018). Building a Sentiment Corpus of Tweets in Brazilian Portuguese. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., e Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

De Choudhury, M., Counts, S., e Horvitz, E. (2013). Social media as a measurement tool of depression in populations. In *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, page 47–56, New York, NY, USA. Association for Computing Machinery.

Duque, J. W. G., Raymundo, A. L., e Neto, P. F. (2018). Uma aplicação de big data para classificação de sentenças depressivas do twitter. *Revista H-TEC Humanidades e Tecnologia*, 2(1):82–95.

Eichstaedt, J. C., Smith, R. J., Merchant, R. M., Ungar, L. H., Crutchley, P., Preoţiuc-Pietro, D., Asch, D. A., e Schwartz, H. A. (2018). Facebook language predicts depression in medical records. volume 115, pages 11203–11208. National Academy of Sciences.

Ji, S., Yu, C., Fung, S.-f., Pan, S., e Long, G. (2018). Supervised learning for suicidal ideation detection in online user content. *Complexity*, 2018:1–10.

Mowery, D. L., Park, A., Bryan, C., e Conway, M. (2016). Towards automatically classifying depressive symptoms from Twitter data for population health. In *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES)*, pages 182–191, Osaka, Japan. The COLING 2016 Organizing Committee.

O'Dea, B., Wan, S., Batterham, P. J., Calear, A. L., Paris, C., e Christensen, H. (2015). Detecting suicidality on twitter. *Internet Interventions*, 2(2):183–188.

Pestian, J., Matykiewicz, P., Linn-Gust, M., South, B., Uzuner, O., Wiebe, J., Cohen, K., Hurdle, J., e Brew, C. (2012). Sentiment analysis of suicide notes: A shared task. *Biomedical informatics insights*, 5:3–16.

Santos, W., Funabashi, A., e Paraboni, I. (2020). Searching brazilian twitter for signs of mental health issues. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6111–6117, Marseille, France. European Language Resources Association.

Souza, F., Nogueira, R., e Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*.

Spitzer, R. L., Kroenke, K., Williams, J. B., Group, P. H. Q. P. C. S., Group, P. H. Q. P. C. S., et al. (1999). Validation and utility of a self-report version of prime-md: the phq primary care study. *Jama*, 282(18):1737–1744.

Yadav, S., Chauhan, J., Sain, J. P., Thirunarayan, K., Sheth, A., e Schumm, J. (2020). Identifying depressive symptoms from tweets: Figurative language enabled multitask learning framework. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 696–709, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yazdavar, A. H., Al-Olimat, H. S., Ebrahimi, M., Bajaj, G., Banerjee, T., Thirunarayan, K., Pathak, J., e Sheth, A. (2017). Semi-supervised approach to monitoring clinical depressive symptoms in social media. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 1191–1198.

Ziwei, B. Y. e Chua, H. N. (2019). An application for classifying depression in tweets. In *Proceedings of the 2nd International Conference on Computing and Big Data*, ICCBD 2019, page 37–41, New York, NY, USA. Association for Computing Machinery.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Detecção de desinformação sobre Covid-19 no Twitter

Ana Alice Ximenes Mota[1], **Wellington Franco**[1] **e César Lincoln Cavalcante Mattos**[1]

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

aliceximenes@alu.ufc.br, wellington@crateus.ufc.br, cesarlincoln@dc.ufc.br

Abstract. *The damage caused by false or misleading news has increased due to the ease with which information is disseminated on social networks. During the Covid-19 pandemic, which began in 2020, such news could generate panic in the population and erroneously instruct people about the prevention of the disease. The present work introduces a new corpus from Twitter posts in the Portuguese language about misinformation from Covid-19*[1]*. In addition to the new corpus, the work evaluates different approaches to textual representations and learning algorithms models in the task of detecting misinformative messages. The best result obtained achieved an F1-score of* 89% *in the SVM classification model with the TF-IDF textual representation.*
Keywords: *Natural Language Processing, misinformation, Covid-19.*

Resumo. *Os danos causados por notícias falsas ou enganosas têm se potencializado graças à facilidade com que as informações são disseminadas em redes sociais. Durante a pandemia do Covid-19, iniciada em 2020, tais notícias foram capazes de gerar pânico na população, além de instruir erroneamente as pessoas sobre a prevenção da doença. O presente trabalho introduz um novo corpus a partir de postagens no Twitter na língua portuguesa com desinformações sobre a Covid-19*[1]*. Além do novo corpus, o trabalho avalia diferentes abordagens de representações textuais e algoritmos de aprendizagem na tarefa de detecção de mensagens contendo desinformação. O melhor resultado obtido alcançou F1-score de* 89% *no modelo de classificação SVM com a representação textual TF-IDF.*
Palavras Chaves: *Processamento de Linguagem Natural, Desinformação, Covid-19.*

1. Introdução

Um estudo publicado em parceria entre *We are social*[2] e *Hootsuite*[3], relatou que em janeiro de 2021 existiam 4,2 bilhões de usuários de redes sociais pelo mundo. Esse número equivale a um crescimento de mais de 13% em relação ao ano de 2020 e a expectativa é que esse aumento continue nos próximos anos [Kemp 2021]. Por consequência, mais pessoas e empresas consomem e expõem informações nas redes sociais. Contudo, essa liberdade acaba permitindo que ocorra uma grande disseminação de informações falsas, comumente chamadas de desinformação ou *fake news*.

[1]Disponível em https://github.com/aliceximenes/fake-news-covid-19.
[2]https://wearesocial.com
[3]https://www.hootsuite.com

De acordo com [Lazer et al. 2018], *fake news* podem ser definidas como informações fabricadas que imitam o conteúdo da mídia de notícias na forma, mas não no processo organizacional ou na intenção. A intenção, por vezes, é manipular a população em diversos contextos. Podemos citar como exemplo de manipulação no contexto político a interferência causada pela *Cambridge Analytica/Facebook* nas eleições presidenciais dos Estados Unidos em 2016 [Confessore 2018].

Com o advento da pandemia da Covid-19, ocorreu o desencadeamento de uma série de informações falsas, atingindo a população como um todo. Por exemplo, algumas desinformações buscam enganar a população indicando meios para prevenir e/ou curar a doença, colocando em risco a sociedade por se tratarem de métodos sem nenhuma comprovação científica.

Visando auxiliar a identificação de desinformações dessa natureza que possam causar malefícios à sociedade, este trabalho propõe a criação de um *corpus* em língua portuguesa sobre a Covid-19. Diversas abordagens de classificação serão avaliadas, incluindo diferentes técnicas de representação textual e algoritmos de aprendizagem de máquina, proporcionando modelos de referência para a tarefa de detecção de desinformação. O novo *corpus*, coletado da rede social Twitter[4], contém postagens com desinformação e não desinformação de cinco tópicos amplamente noticiados sobre o novo Coronavírus, causador da Covid-19.

O presente artigo é organizado da seguinte forma: na próxima seção, é feito um breve resumo dos trabalhos relacionados; na Seção 3 relata-se detalhadamente a metodologia usada para a construção do corpus, pré-processamento dos dados e modelagem da solução; na Seção 4 é feita a explicação de como os experimentos foram realizados e os resultados obtidos são discutidos; por fim na Seção 5, os resultados encontrados são discutidos e direções para investigações futuras são apontadas.

2. Trabalhos relacionados

Em [Monteiro et al. 2018], foi proposto o primeiro *corpus* na língua portuguesa com a finalidade de detectar notícias falsas. O objetivo do trabalho foi criar o *corpus*, Fake.Br, com 3100 notícias verdadeiras e 3100 notícias falsas. As notícias foram coletadas em sites e blogs, sendo as classificadas como verdadeiras extraídas de grandes veículos de notícias, como G1, Folha de São Paulo e Estadão. Em um trabalho posterior, diversas representações textuais e modelos de aprendizagem de máquina foram explorados e avaliados no mesmo *corpus* [Silva et al. 2020].

A criação do Fake.Br abriu caminho para múltiplas pesquisas científicas e para a construção de novos *corpora* sobre desinformação em língua portuguesa. Muitas dessas focam na coleta e experimentação de dados extraídos de aplicativos de mensagens e de redes sociais. Em [Cabral et al. 2021] é feita a criação do *corpus* FakeWhastApp.BR a partir de dados coletados em grupos públicos de conversas no aplicativo WhatsApp[5]. Em relação a redes sociais, uma amplamente usada e foco de pesquisas atuais é o Twitter.

Com uma projeção para o ano de 2021 de 322.4 milhões de usuários [Newberry 2021], o Twitter vem sendo alvo de diversas pesquisas acadêmicas. Em

[4]https://twitter.com
[5]https://www.whatsapp.com/

[Cordeiro and Pinheiro 2019] é feita a construção do *corpus* intitulado FakeTweet.Br com dados coletados do Twitter em português. Por se tratar de postagens que possuem limite de tamanho (280 caracteres), o *corpus* FakeTweet.Br proporciona experimentos em textos com formatos diferentes dos encontrados em blogs, sites e aplicativos de mensagens.

Na língua inglesa, em [Buntain and Golbeck 2017], foi construído um sistema automatizado para detectar desinformações em tópicos populares do Twitter. Foram utilizados dois conjuntos de dados para o aprendizado: CHEDBANK, introduzido por [Mitra and Gilbert 2015], constituindo-se de uma base de dados *crowdsourced* em que os tópicos contidos nos tweets estão identificados; e PHENE, um conjunto de dados de rumores em potencial do Twitter [Zubiaga et al. 2016]. Por fim, o método desenvolvido foi aplicado a uma base de dados de notícias falsas.

O uso de mecanismos de identificação de desinformações nos dados do Twitter também foi utilizado por [Zervopoulos et al. 2020] nos protestos políticos ocorridos em Hong Kong. Foi utilizado um conjunto inicial de postagens falsas em inglês e chinês, sendo este último traduzido para o inglês na sequência.

Uma visão geral dos trabalhos citados está apresentada na Tabela 1. Nela, são indicados o título e ano dos artigos, o idioma do *corpus*, a fonte de dados usada para sua construção e a informação se o rótulo do *corpus* foi construído de forma manual, semi-automático ou automático.

Até o conhecimento dos autores, não existia um corpus público em português obtido através de dados do Twitter com desinformações sobre a Covid-19. Diante disso, o presente trabalho, tem o intuito de construir um *corpus* com essas especificações, além de realizar experimentos e comparar os seus resultados.

Tabela 1. Resumo dos trabalhos relacionados.

Título	Idioma	Fonte	Rótulo	Ano
Automatically identifying fake news-sin popular twitter threads	Inglês	Sites e Twitter	Semi-automático	2017
Contributions to the study of fake news in portuguese: New corpus and automatic detection results	Português	Blogs e sites	Semi-automático	2018
Um corpus de notícias falsas do twitter e verificação automática de rumores em língua portuguesa.	Português	Twitter	Manual	2019
Towards automatically filtering fake news in portuguese	Português	Blogs e sites	Semi-automático	2020
Hong Kong Protests: Using Natural Language Processing for Fake News Detection on Twitter	Inglês	Twitter	Automático	2020
FakeWhastApp.BR: NLP and Machine Learning Techniques for Misinformation Detection in Brazilian Portuguese WhatsApp Messages	Português	WhatsApp	Manual	2021

3. Metodologia

Este trabalho propõe a criação de um novo *corpus* contendo postagens do Twitter sobre determinados tópicos falsos da Covid-19. A Figura 1 retrata as etapas realizados no fluxo de construção do novo *corpus*. Essas etapas serão detalhadas na sub-seção seguinte.

Em adição ao novo *corpus*, foi realizado um extenso trabalho de experimentação de técnicas textuais e modelos de classificação para encontrar a abordagem mais assertiva na classificação de tweets entre desinformação e não desinformação.

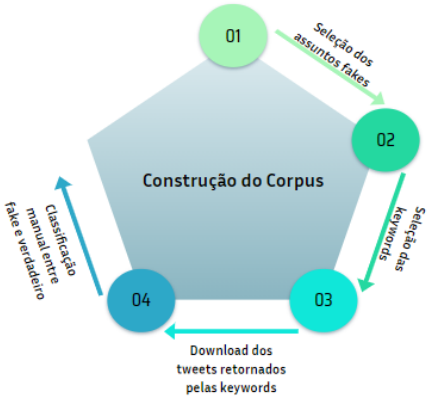


Figura 1. Fluxo de criação do corpus.

3.1. Construção do Corpus

Em meados de 2008 foi criado o primeiro *corpus* sobre desinformações na língua portuguesa [Monteiro et al. 2018] e como, até o conhecimento dos autores, não existia um *corpus* público em português, rotulado e proveniente do Twitter com dados de desinformações sobre a Covid-19. Foi necessário construir um novo *corpus* para a realização de um estudo sobre o tema.

Inicialmente, foram escolhidos tópicos falsos amplamente divulgados como métodos de tratamento e/ou prevenção eficazes contra a doença. Os tópicos estão relacionados na Tabela 2 e foram retirados do site do Governo Federal Brasileiro[6] criado para combate das desinformações sobre a Covid-19.

Tabela 2. Tópicos com desinformações usadas para coleta no Twitter.

	Tópicos
1	Chá de limão com bicarbonato quente cura coronavírus
2	Beber muita água e fazer gargarejo com água morna, sal e vinagre previne coronavírus
3	Vitamina C + zinco e o novo coronavírus
4	Ivermectina tem eficácia comprovada contra a Covid
5	Beber água quente mata o coronavírus

[6]https://antigo.saude.gov.br/fakenews/

Após a escolha dos tópicos, seguiu-se para a etapa de *web crawler* dos tweets. O processo foi realizado filtrando os tweets com as palavras chaves dos tópicos da Tabela 2 e palavras relacionadas ao novo Coronavírus. A coleta ocorreu entre fevereiro e setembro de 2020.

Posteriormente, o processo de rotulação foi iniciado. Foi realizada a leitura de cada tweet coletado e atribuído o rótulo de desinformação ou não. Para os casos que não se enquadravam nessas categorias, por exemplo por terem um viés de sarcasmo ou não serem relacionados ao assunto, o tweet não foi adicionado ao *corpus*. Também foram observados diversos tweets repetidos, os quais não foram considerados. Por fim, o *corpus* construído possui 730 tweets, sendo 456 rotulados como desinformação e 274 como não desinformação. A Tabela 3 apresenta alguns exemplos.

Tabela 3. Amostra do *corpus* criado.

Tweets	**Classificação**
O coronavírus pode ser curado se você fizer gargarejo com bicarbonato de sódio e limão.	Desinformação
Receita com limão e bicarbonato, além de não evitar mortes por coronavírus, pode ser prejudicial à saúde.	Não desinformação

3.2. Pré-processamento dos Dados

Com a construção do *corpus*, a etapa seguinte foi de tratamento e limpeza dos dados. As bibliotecas NLTK[7] e *Regex*[8] do Python foram utilizadas nessa etapa. O primeiro passo foi colocar todos os tweets em letras minúsculas. Além disso, foram retirados os acentos, pontuações, números, palavras de parada (*stopwords*) e eventuais links que direcionavam para outros sites. Em seguida, foi realizado o processo de tokenização, que consiste em dividir os textos em segmentos demarcados de caracteres, chamados *tokens*.

Por fim, experimentamos algumas técnicas de codificação textual amplamente usadas na literatura. Foram testadas as seguintes técnicas: o tradicional método de *Bag of Words* (BoW) e Frequência do Termo – Frequência Inversa dos Documentos (TFIDF); e o método de representação textual Word2Vec [Mikolov et al. 2013].

As codificações BoW e TFIDF padrão tratam cada palavra de forma individual, descartando inteiramente a ordem de aparecimento no texto. A técnica *ngram* permite preservar algumas dessas informações. Essa abordagem permite considerar uma sequência de palavras adjacentes como um único termo. Com isso, a sequência pode ser composta por uma palavra (forma padrão do BoW e TFIDF) ou *ngramas*, em que *n* é a quantidade de palavras adjacentes.

Dessa forma, para as representações BoW e TFIDF foram testados diferentes valores para o hiperparâmetro *ngram_range* (1gram, 2gram ou 3gram) e foi definido a remoção de termos com uma frequência menor ou igual a 1% nos tweets coletados. Na representação Word2Vec, usou-se os vetores de palavras pré-treinados propostos em [Hartmann et al. 2017]. Os modelos utilizados para gerar esses vetores foram treinados

[7]https://www.nltk.org/

[8]https://pypi.org/project/regex/

com documentos da língua portuguesa de 17 conjuntos de dados de diferentes domínios, totalizando 1.395.926.282 *tokens*. Foram considerados vetores com 300 dimensões treinados com a abordagem *Skip-Gram*, pois foi a abordagem com o melhor resultado experimental obtido da técnica.

3.3. Modelos de Aprendizagem de Máquina e Métricas de Avaliação

Tendo finalizado o pré-processamento dos dados, iniciou-se o preparo do *corpus* para a etapa de aprendizagem. Para tanto, dividiu-se a base de dados aleatoriamente em 70% para treino e 30% para teste. A última parcela foi usada exclusivamente para avaliar a capacidade de generalização dos algoritmos de aprendizagem.

Os modelos de classificação binária comparados foram: LightGBM (LGBM), Máquina de Vetores de Suporte (*Support Vector Machines*, SVM), *Random Forest* (RF), *Naive Bayes* (NB), AdaBoost (AB) e Regressão Logística. Para o LGBM, foi usada a implementação original[9], para os demais modelos foram usadas as implementações da biblioteca scikit-learn [Pedregosa et al. 2011][10].

Para cada modelo de aprendizagem avaliado foi realizado um processo de escolha dos melhores hiperparâmetros usando o método de pesquisa em grade (*grid search*), implementado na biblioteca scikit-learn. Para validar o resultado do modelo em cada combinação de hiperparâmetros foi realizado, na base de treino, o cálculo da métrica F1-score e o método de validação cruzada em *k-folds*, sendo $k = 5$.

As métricas consideradas para a avaliação final dos classificadores, nos dados reservados para teste, foram as seguintes: acurácia, indica a performance geral do modelo, ou seja, a fração de quantos tweets com desinformação foram preditos como desinformação; precisão, a fração dentre todas as classificações de tweets em desinformação pelo modelo, quantas estão corretas; revocação, dentre todos os casos de tweets com desinformação qual a fração de acerto; F1-score é a média harmônica entre precisão e revocação; e a área sob a curva ROC, indica a capacidade discriminativa do modelo, isto é, a capacidade de classificar corretamente tweets em desinformação e não desinformação.

4. Experimentos e Discussão de Resultados

Considerando-se a parcela de dados reservada para treinamento (70% do *corpus*), a etapa de *grid search* para a escolha da melhor combinação de hiperparâmetros dos modelos envolveu também a avaliação de diferentes *pipelines* de processamento do *corpus*. As *pipelines* consideradas testaram as codificações BoW e TFIDF com diferentes valores para o hiperparâmetro *ngram*, além de avaliar a representação textual Word2Vec. A Tabela 4, apresenta os hiperparâmetros testados em cada modelo. Ressalta-se que os códigos dos experimentos e o *corpus* construído estão disponíveis em repositório público.

Os resultados obtidos com as representações BoW e TFIDF aplicadas na base de teste, estão apresentados na Tabela 5. Destaca-se que os modelos obtidos proporcionam soluções de referência para pesquisas futuras. Nota-se que em todos os modelos as métricas foram, em sua maioria, acima de 80%, indicando bons resultados de detecção.

[9]https://lightgbm.readthedocs.io/
[10]https://scikit-learn.org/

Tabela 4. Hiperparâmetros testados em cada modelo.

Modelos	Hiperparâmetros
LGBM	Número máximo de folhas em uma árvore e regularização L1 e L2
SVM	Função de kernel e gamma dependendo da função de kernel
Random Forest	Número de estimadores e índice de pureza
Naive Bayes	Parâmetro de suavização aditivo
AdaBoost	Número de estimadores, taxa de aprendizagem e algoritmo utilizado
Reg. Logística	Tipo de penalização e o algoritmo utilizado

Tabela 5. Melhores resultados de cada modelo (BoW e TF-IDF).

Modelo	Acurácia	Precisão	Revocação	F1-Score
LGBM Classifier	0,78	0,80	0,83	0,82
SVM	**0,85**	**0,82**	**0,97**	**0,89**
Random Forest	0,84	0,83	0,91	0,87
Naive Bayes	0,79	0,87	0,77	0,81
AdaBoost	0,84	0,86	0,89	0,87
Reg. Logística	0,82	0,79	0,95	0,86

Na Figura 2, temos a curva ROC e a área sob a curva ROC (AUC, *area under the curve*) indicando bons resultados em todos os modelos, em especial o Máquina de Vetores de Suporte (SVM). A métrica usada para escolher o melhor modelo foi o F1-score. Logo, o melhor modelo obtido nesse experimento foi o Máquina de Vetores de Suporte com 89% de F1-score. Para esse modelo, a melhor *pipeline* e conjunto de hiperparâmetros encontrados foi utilizando a codificação TF-IDF, *1gram*, usando a função de kernel RBF e o *gamma scale*.

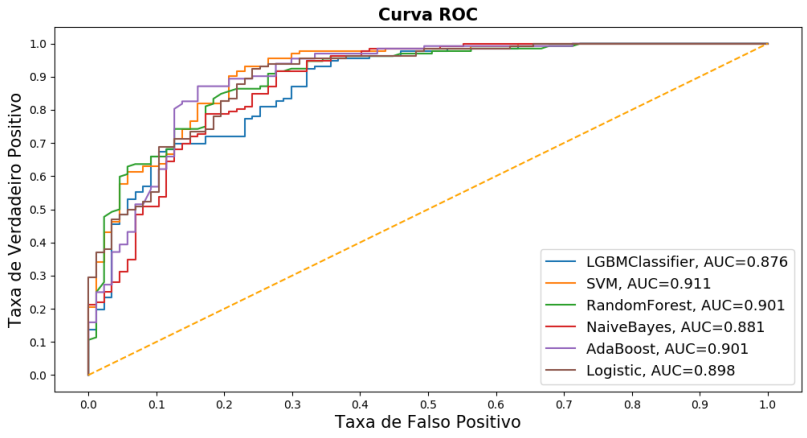


Figura 2. Curva ROC dos experimentos com BoW e TFIDF.

Na Tabela 6 e Figura 3 estão os resultados dos experimentos com a técnica Word2Vec. O desempenho foi abaixo do encontrado na abordagem anterior. O comporta-

mento médio das métricas foi em torno de 70% e não houve um modelo que se sobressaiu aos demais. Entretanto, entre todos os modelos, o LGBM e o SVM foram os que obtiveram maior F1-score, cerca de 76%. O LGBM obteve o melhor desempenho, com 76, 8% de F1-score. Para esse modelo, o conjunto de hiperparâmetros escolhido foi o valor 5 para o número máximo de folhas, a regularização L1 com termo 0,1 e sem regularização L2 (termo igual a 0).

Tabela 6. Melhores resultados de cada modelo com Word2Vec.

Modelo	Acurácia	Precisão	Revocação	F1-Score
LGBMClassifier	**0,69**	**0,70**	**0,84**	**0,76**
SVM	0,61	0,61	1,00	0,75
RandomForest	0,66	0,66	0,88	0,76
AdaBoost	0,67	0,69	0,81	0,75
Reg. Logística	0,62	0,62	0,92	0,74

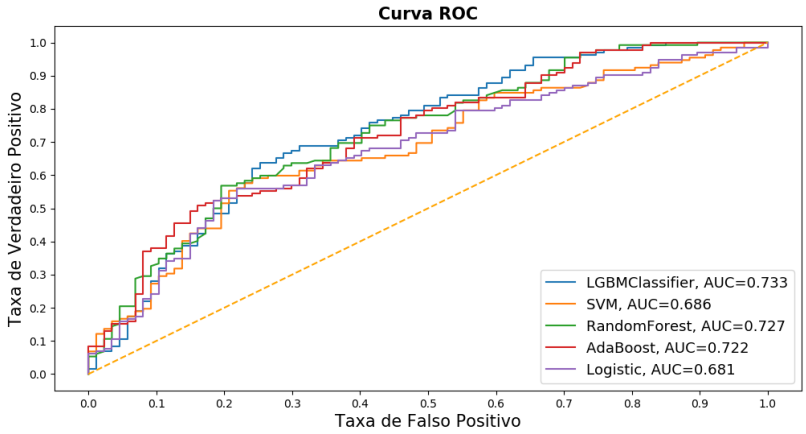


Figura 3. Curva ROC dos experimentos com Word2Vec.

5. Conclusão

O presente trabalho propôs a construção de um *corpus* coletado do Twitter com desinformações relacionadas à Covid-19. O novo *corpus* foi usado em experimentos que avaliaram diferentes representações textuais e modelos de aprendizagem de máquina para classificar tweets em desinformação ou não desinformação. Os resultados indicaram o modelo SVM com representação textual TFIDF como a melhor abordagem, com F1-score de 89% usando *1gram* e função de *kernel* RBF.

Trabalhos futuros envolvem a análise de erros dos classificadores. Identificando se os modelos erram ou não os mesmos tweets, se os tweets mais difíceis de classificar corretamente possuem pontos em comum, se um modelo de *ensemble* com os classificadores que cometem erros distintos resulta em um melhor resultado. Outra vertente consiste em investigar questões éticas relacionadas a eventuais vieses na classificação.

Referências

Buntain, C. and Golbeck, J. (2017). Automatically identifying fake news in popular twitter threads. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 208–215. IEEE.

Cabral, L., Monteiro, J. M., da Silva, J. W. F., Mattos, C. L. C., and Mourao, P. J. C. (2021). FakeWhastApp.BR: NLP and machine learning techniques for misinformation detection in brazilian portuguese whatsapp messages. In *Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 63–74. INSTICC, SciTePress.

Confessore, N. (2018). Cambridge analytica and facebook: The scandal and the fallout so far. `https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html`. Acessado em : 20/07/2021.

Cordeiro, P. R. and Pinheiro, V. (2019). Um corpus de notícias falsas do twitter e verificação automática de rumores em lingua portuguesa. In *STIL-Brazilian Symposium in Information and Human Language Technology. IEEE, Salvaldor, BA, Brazil*, pages 220–228.

Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Rodrigues, J., and Aluisio, S. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*.

Kemp, S. (2021). Digital 2021: the latest insights into the 'state of digital'. `https://wearesocial.com/blog/2021/01/digital-2021-the-latest-insights-into-the-state-of-digital`. Acessado em : 20/07/2021.

Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., et al. (2018). The science of fake news. *Science*, 359(6380):1094–1096.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mitra, T. and Gilbert, E. (2015). Credbank: A large-scale social media corpus with associated credibility annotations. In *Ninth international AAAI conference on web and social media*.

Monteiro, R. A., Santos, R. L., Pardo, T. A., De Almeida, T. A., Ruiz, E. E., and Vale, O. A. (2018). Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In *International Conference on Computational Processing of the Portuguese Language*, pages 324–334. Springer.

Newberry, C. (2021). 36 twitter statistics all marketers should know in 2021. `https://blog.hootsuite.com/twitter-statistics/`. Acessado em : 20/07/2021.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Silva, R. M., Santos, R. L., Almeida, T. A., and Pardo, T. A. (2020). Towards automatically filtering fake news in portuguese. *Expert Systems with Applications*, 146:113199.

Zervopoulos, A., Alvanou, A. G., Bezas, K., Papamichail, A., Maragoudakis, M., and Kermanidis, K. (2020). Hong kong protests: using natural language processing for fake news detection on twitter. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 408–419. Springer.

Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., and Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

A Long Texts Summarization Approach to Scientific Articles

Cinthia M. Souza[1], **Renato Vimieiro**[1]

[1]Universidade Federal de Minas Gerais, Brazil

{cinthiasouza, rvimieiro}@dcc.ufmg.br

Abstract. *Automatic text summarization aims at condensing the contents of a text into a simple and descriptive summary. Summarization techniques drastically benefited from the recent advances in Deep Learning. Nevertheless, these techniques are still unable to properly deal with long texts. In this work, we investigate whether the combination of summaries extracted from multiple sections of long scientific texts may enhance the quality of the summary for the whole document. We conduct experiments on a real world corpus to assess the effectiveness of our proposal. The results show that our multi-section proposal is as good as summaries generated using the entire text as input and twice as good as single section.*

1. Introduction

In this work we propose an alternate approach for summarizing long scientific texts. We investigate whether the combination of summaries extracted from multiple sections may enhance the quality of the summary for the whole document. This could be of particular interest for summarization methods that are unable to deal with long texts due to size restrictions, particularly those with high computing demands.

Automatic text summarization aims at facilitating access to information. The objective of this technique is to condense the information in a text into a simple and descriptive summary, which gives the reader a general idea of the text without having to read its entire content. Text summarization techniques can be divided into two groups: extractive; and abstractive. Extractive Text Summarization (ETS) techniques assign a score to each sentence of the text and select n sentences with the highest score to compose the summary. Abstractive Text Summarization (ATS) techniques are trained to generate a natural language summary using an internal representation of the text. Because they have a large dictionary, these techniques do not necessarily create summaries with the same words as the text. This makes the summarization process more similar to humans'. For a long time, ETS techniques used simpler models of unsupervised learning and ATS techniques were little explored compared to ETS. In recent years, ETS and ATS techniques have had great improvements due to the use of Deep Learning (DL) models.

Most of recent works explore text summarization in news corpora such as CNN-Daily/Mail, DUC Corpus and Gigaword. These corpora are mostly comprised of short texts with approximately 650 words [Nallapati et al., 2016]. In short texts, DL models are easily applicable and perform well compared to unsupervised techniques. However, DL models present some challenges regarding their training, such as the difficulty of working with long texts, due to the high computational cost [Ding et al., 2020]. Solutions to this problem are to use sliding windows, simplify the architecture, or set a maximum number of words that will be used as input. There are different negative implications attached to

using each of these strategies. One of them is the loss of important information. In the task of scientific text summarization, for example, the abstracts of the articles, which are used as ground-truth, are created by experts, and follow a standardized structure, in which there is a contextualization, the problem, the solution and the evaluation of results. This content is commonly distributed in different sections of the text. So not considering all the information in the text reduces the solution space, resulting in a loss of performance. We compare five ETS algorithms in a long text dataset[1], composed of scientific articles, published by the Plos One journal. Our main contributions are:

- Assessing the performance of ETS methods on a long text corpus;
- Exploring the segmentation of scientific articles considering the structural pattern of scientific abstracts;
- Evaluating the contribution of the sections in the generation of summaries;
- Evaluating the impact of combining multi-section summaries on the final performance of algorithms;
- Validating the results using the set of metrics Recall-Oriented Understudy for Gisting Evaluation[2] (ROUGE) [Lin, 2004].

The remaining of this manuscript is organized as follows. In Section 2 we present a perspective of how the summarization task has been explored. Then, in Section 3 we describe the algorithms used in this work. Sections 4 and 5 contain the proposed approach with a description of the corpus and the methodological steps and results obtained, respectively. Finally, we conclude the manuscript presenting our final remarks and future works in Section 6.

2. Related Works

ETS techniques are more suitable to particular tasks than ATS techniques. For instance, in situations where it is mandatory to have full control of the content present in the summaries, like in the summarization of scientific and legal documents, ETS is more apropriate since changing or introducing more sentences to the summary may alter the document meaning. We discuss in this section some recent works in the area of ETS. These works were chosen in order to present a perspective of how the ETS task is currently being explored.

Gidiotis and Tsoumakas [2020] proposed in their work a divide-and-conquer approach for long text summarization. The proposed approach uses discourse structure and sentence similarity to create a dataset composed of pairs of short texts and their summaries. The aim of the authors is to reduce the complexity of the problem, consequently, reducing the computational cost. The proposed method was tested on different summarization algorithms, including SumBasic [Vanderwende et al., 2007], LexRank [Erkan and Radev, 2004], and PEGASUS [Zhang et al., 2020]. According to the authors, the best models obtained presented results that were competitive with the state-of-the-art (SOTA). Dong et al. [2021] proposes an unsupervised ETS model for long scientific texts based on graphs. Their approach works on a hierarchical graph representing the document. The hierarchy has two levels of connection, intra-section and inter-section. The similarity between sentences is calculated using the cosine-similarity and the importance of each

[1]Code and dataset available at: https://github.com/CinthiaS/long_text_summarization
[2]Available at: https://github.com/chakki-works/sumeval

sentence is the sum of the intra and inter-section importance. The intra-section is the comparison of sentences within the same section. The inter-section is obtained by comparing the sentences of a section with those of other topics/sections of the document. The summaries are created by extracting the sentences with the highest score. The experiments were conducted using PubMed and ArXiv datasets. The results were compared with supervised and unsupervised models, in addition to the baseline lead, which selects the first k tokens as the summary, and an Oracle. To validate the results, the metrics ROUGE-1, ROUGE-2, ROUGE-L and the human evaluation were used. The results obtained were better than all the compared algorithms.

Among the works studied, only Gidiotis and Tsoumakas [2020] explores text segmentation as a strategy for long text summarization. The results obtained by the authors show that the summarization of sections of the text, individually, and the creation of summaries as the composition of these summaries is an efficient strategy for long text summarization. Differently from the proposal of this work, Gidiotis and Tsoumakas [2020] propose an agnostic strategy to the knowledge domain of the text. However, we believe that performing a segmentation considering the abstract structure of articles can achieve better results. Furthermore, based on the works of Xu et al. [2019], Zhong et al. [2020], Xiao et al. [2020], and Zhang et al. [2020], which are important references in the field, it is possible to see that these, in general, focus on news corpora, which are characterized by short text and summaries. Thus, there is a need to explore strategies that can allow the use of reference models in the area in this scenario.

3. Extractive Summarization Algorithms

In this work, five ETS algorithms are tested: SumBasic; LexRank; TextRank; and two models based on Bidirectional Encoder Representations from Transformers (BERT). SumBasic was selected as the baseline method for comparison. LexRank and TextRank are graph-based algorithms that stand out for having a simple approach with competitive results. The last two algorithms are approaches that use latest SOTA natural language processing components, which are language models created with the BERT architecture. SumBasic [Vanderwende et al., 2007] evaluates the importance of the words in the text based on their frequencies. After, it assigns an importance to the sentences of the text according to the importance of their words [Vanderwende et al., 2007]. The idea is that the more important words a sentence has, the more important it is. LexRank [Erkan and Radev, 2004] and TextRank [Mihalcea and Tarau, 2004] are graph-based algorithms. Basically, these algorithms create a representation of the text in a weighted undirected graph, where vertices are sentences, edges represent the relationship between two sentences, and edge weights are the similarity between them. The main differences between these algorithms is the calculation of similarity. LexRank defines the similarity between two sentences as the cosine of their Term Frequency–Inverse Document Frequency (TF-IDF) vector representations, while TextRank uses a measure of overlap between the words in the text, normalized by the length of the sentences. After creating the text representation, both algorithms use the PageRank algorithm [Page et al., 1999] to assign a score to the sentences. The summary is composed of the k sentences with the highest score.

The BERT Summarizer algorithm is based on clustering and uses representations of embeddings generated by the BERT model. BERT is a pre-trained Transformer architecture [Vaswani et al., 2017], designed for creating deep representations of unlabeled

text. One of the advantages of BERT is that the architecture used is bidirectional, making it possible to associate forward and backward contexts for all layers, unlike, for example, the Generative Pre-trained Transformer (OpenAI GPT), which is unidirectional [Devlin et al., 2018]. Devlin et al. [2018] describes BERT as being conceptually simple and empirically powerful. Basically, the BERT Summarizer obtains the embedding representations of the text sentences using BERT and generates a matrix where lines represent sentences and columns represent the dimensionality of the embedding vector. This matrix is used as input for the K-means algorithm, together with the number k of clusters which also represents the number of sentences to be extracted. At the end, the sentence that has the smallest distance from the centroid is added to the summary.

4. Proposed Approach

In this work we use a corpus of scientific articles published by Plos One. The corpus is publicly available from the journal's website[3]. The collected articles were segmented considering their division of sections. To retrieve these sections, it is important to use a tagged base that allows the recognition of these sections. The data provided by Plos One is made available in XML, allowing easy recognition of sections. The experiments are carried out in a dataset with 5000 articles, selected at random. Each document is segmented into four sections. The segmentation of the dataset is used as a strategy to work with long texts, reducing the amount of data in the input of the algorithm and enabling the capture of information from each section in order to generate comprehensive summaries, covering each topic of the abstract. Table 1 presents the name of the sections extracted from the articles, the acronyms used to identify them, the average number of sentences and words in each of them and the compression ratio, which is the ratio between the number of sentences in the section and the number of sentences in the abstract.

Table 1. Description of the corpus used in the experiments

Section	Acronyms	Average number of sentences	Average number of words	Compression
Abstract	S_1	11	210	-
Introduction	S_2	23	540	2,09
Materials	S_3	59	1077	5,36
Results and Conclusion	S_4	110	2081	10,00
All document	D_m	192	3698	17,45

Our approach is divided into three steps. In the first step, data collection and preprocessing is performed. The dataset creation process consists of three phases: collection, refactoring and segmentation. Initially, the documents are collected in XML format. In some cases, the XML document did not have a tag delimiting the text sections. By consequence, the collected documents were refactored in order to correct inconsistencies and facilitate the segmentation process. The refactoring process generated a new XML document where all target sections are properly tagged. The documents are segmented afterwards into four sections, S_1, S_2, S_3, S_4. The section S_1 is the abstract of the article, used as ground-truth, henceforth called the reference summary. The preprocessing step follows the document segmentation. The first task in this step is removal of text citations and section titles. After, the XML is converted to text and noise, that is special characters,

[3]Available at: http://api.plos.org/text-and-data-mining/ - Accessed on: Aug 2021

excess spaces and line breaks, and unicode symbols are removed. Finally, all texts are lowercased, stop words are removed, and the words remaining are stemmed.

In the second step, the texts are summarized using the five algorithms described in Section 3. Initially, the experiments are performed by section. Each algorithm receives the text of each section, separately, and generates a summary of each. The algorithms used receive the number k of sentences to be extracted. We define that the summaries of each section are generated with $k = 3$, so the composition of the summaries will have 9 sentences which is, approximately, the average number of sentences in an abstract (see Table 1). Subsequently, a summary is generated from the entire content of the text. The objective of these experiments is: (1) evaluate the contribution of each section to the summary and (2) evaluate whether the summaries created in each section, separately, have as good results as using the entire text as input. Two summary generation approaches were used. The first, called A_1, uses all the text as input to the algorithms. The second, called A_2, creates a summary for each section and combines the summaries. A_1 uses $k = 9$ and A_2 uses $k = 3$, thus, A_2 generates a summary with 9 sentences, 3 from each section. Thus, both are limited to the same summary size. The SumBasic, LexRank and TextRank algorithms are unsupervised, so they do not require training. Models using BERT were implemented with an API[4] developed by Miller [2019]. The difference between the two algorithms is that, the first one, called BERT Basic uses a generic pre-trained model, provided by the organization Hugging Face[5]. The second one, called SciBERT Summ, uses a pre-trained model created from scientific texts, provided by the Allen Institute for AI, called SciBERT[6].

Finally, the third step is the evaluation of the generated summaries. For this, the metrics ROUGE-1 (R1), ROUGE-2 (R2) and ROUGE-L (RL) are used. The ROUGE metrics is widely used in the literature. ROUGE-N evaluates the overlap of n-grams between the candidate and reference summary [Sanchez-Gomez et al., 2018]. ROUGE-L evaluates the correspondence between the Longest Common Substring (LCS) shared by two sentences [Sanchez-Gomez et al., 2018]. Both metrics assign a score from 0 to 1 to each sentence, where 0 represents that the candidate summary does not capture any information from the reference summary and 1 represents that the candidate summary captures all information of the reference summary. The results obtained with the summarization algorithms, which we called candidate summary, are compared with the reference summaries using ROUGE metrics. The performance of an algorithm is calculated as the average of the metrics for each generated summary.

5. Results

Initially, the summary of each section of the text was performed using the algorithms presented in Section 4. For each section, 3 sentences were extracted. The results obtained were compared with the reference summary and are presented in Table 2a. The first column presents the name of the algorithm used, the second presents the acronyms of the section (see Table 1). The columns R1, R2 and RL show the results of the metrics and, the last column, presents the average of the three metrics. The algorithm that presented the best result is highlighted.

[4]https://github.com/dmmiller612/bert-extractive-summarizer
[5]https://github.com/huggingface/transformers
[6]https://github.com/allenai/scibert

Table 2. Results of ETS experiments conducted in the Introduction (S_2), Materials and Methods (S_3), and Results and Conclusion (S_4) sections and using as input all text (A_1) and using a summary created from the combination of multi-section summaries (A_2).

(a) Results by section

		R1 (%)	R2 (%)	RL (%)	Average
SumBasic	S_2	15.24	4.78	10.52	10.2
	S_3	10.30	3.64	7.89	7.3
	S_4	18.25	6.67	12.68	12.5
LexRank	S_2	17.36	5.55	11.69	11.5
	S_3	13.10	4.84	9.78	9.2
	S_4	25.60	10.01	16.75	17.5
TextRank	S_2	19.76	6.23	12.92	**13.0**
	S_3	15.35	5.51	11.02	**10.6**
	S_4	27.78	10.57	17.74	**18.7**
BERT Basic	S_2	15.45	5.12	10.94	10.5
	S_3	10.91	4.61	8.69	8.1
	S_4	20.21	8.27	14.46	14.3
SciBERT Summ	S_2	15.57	5.16	11.00	10.6
	S_3	11.29	4.69	8.91	8.3
	S_4	21.06	8.68	14.93	14.9

(b) Results by approach

		R1 (%)	R2 (%)	RL (%)	Average
SumBasic	A_1	31.62	10.21	17.27	19.7
	A_2	28.15	10.18	17.37	18.6
LexRank	A_1	37.36	14.44	20.69	**24.2**
	A_2	33.22	12.88	20.05	**22.1**
TextRank	A_1	33.17	13.09	18.80	21.7
	A_2	33.01	12.88	19.69	21.9
BERT Basic	A_1	30.70	10.07	17.34	19.37
	A_2	29.39	11.97	18.97	20.1
SciBERT Summ	A_1	31.53	10.82	17.59	19.98
	A_2	29.79	12.17	19.07	20.3

By comparing the scores present in Table 2a, we concluded that the worst results were obtained using the section Materials and Methods (S_3) and the best ones were using Results and Conclusion (S_4). For the TextRank algorithm, for example, which showed the best performance, the difference between the average of the metrics for the results with Introduction (S_2) was 5.7% and 8.1% for S_3. In all cases, the average results of BERT Basic and SciBERT Summ were worse compared to TextRank and LexRank, being superior only to the baseline, SumBasic. Thus, we concluded that the text sections present different degrees of contribution to the summary generation. The choice of the sections in which the experiments are performed can generate significant changes in the results of the algorithms. Furthermore, it is possible to verify that, although S_2 and S_3 have a lower performance than S_4, there is a contribution of these sections in the summary. After, was performed a comparison between two summarization approaches. Table 2b presents the results obtained with these experiments. The first column presents the algorithms used, the second column identifies the approach used, the columns R1, R2 and RL present the values of the metrics in percentage and, the last column, presents the average of the three metrics.

From Table 2b, it is possible to verify that the results using the combination of summaries is twice as good that summaries of only one section. Thus, it can be concluded that the combination of multi-section summaries is capable of producing high quality summaries. The difference between the approaches is 1.10% for SumBasic, 2.10% for LexRank, 0.2% for TextRank, for BERT Basic, 0.73% and 0.32% for SciBERT Summ. Based on the results, we conclude that segmenting the texts, considering the structural pattern of the abstract, and summarize each section and combine them is a strategy that can help in the task of long texts summarization, reducing the computational cost of algorithms and can mitigate the loss of information. Among algorithms, the best result was obtained with LexRank, for both approaches. The difference between the averages of LexRank with A_1 for the other algorithms were 4.5% for SumBasic, 2.5% for TextRank, 4.38% for BERT Basic, and 4.22% for SciBERT Summ. For A_2 the differences were 3.5% for SumBasic, 0.2% for TextRank, 2.0% for BERT Basic, and 1.8% for SciBERT Summ. TextRank presented the smallest percentage difference between the approaches.

Showing that the summaries of both approaches are very similar. The SumBasic algorithm, which is used as a baseline, had the worst performance. BERT based algorithms showed a worse performance compared to LexRank and TextRank. Even though TextRank has the smallest difference between the approaches, the average metric value of A_1 and A_2 shows that LexRank performs better, with a difference of 2.5% with A_1 and 0.2% with A_2.

6. Final Considerations

Currently, text summarization has shown significant improvements due to the use of DL techniques. However, in this context, long texts summarization is still challenging. In this work, the contribution of different sections of the text in the composition of summaries of scientific articles are evaluated. The results obtained show that the text sections have different degrees of contribution in the generation of summaries. In this experiment, the algorithm with the best performance was TextRank, with a mean of the metrics of 13% for Introduction, 10.6% for Materials and Methods, and 18.7% for Results and Conclusion. Considering all algorithms the best results were obtained with the Results and Conclusion section. When comparing approaches A_1 and A_2, the biggest difference was 2.1% and a lowest was 0.2%. This demonstrated that the combination of multi-section summaries can generate summaries of similar quality compared to using the entire text as input. In this experiment, the best performance was obtained with the LexRank algorithm, with an average score of 22.1% for the proposed approach. Although the results of the metrics obtained in this work are inferior to the SOTA, we believe that the results obtained are promising, as it was conducted in a reduced dataset, using simple unsupervised algorithms and pre-training language representations without no adjustment to the knowledge domain in which it was applied. For future work, we intend to reproduce the experiments in a larger dataset, develop a specific solution for scientific texts and evaluate the results using metrics that allow evaluating using other metrics. The ROUGEs metrics are widely used in the literature, however works such as Souza et al. [2021] and Kane et al. [2020] question the performance of these metrics and highlight the need to explore other metrics.

Acknowledgements

We would like to thank the partial support from MPMG through the project Analytical Capabilities.

References

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *The 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, 2018.

M. Ding, C. Zhou, H. Yang, and J. Tang. Cogltx: Applying bert to long texts. *Advances in Neural Information Processing Systems*, 33, 2020.

Y. Dong, A. M. Romascanu, and J. C. K. Cheung. Discourse-aware unsupervised summarization for long scientific documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1089–1102, 2021.

G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

A. Gidiotis and G. Tsoumakas. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040, 2020.

H. Kane, M. Y. Kocyigit, A. Abdalla, P. Ajanoh, and M. Coulibali. Nubia: Neural based interchangeability assessor for text generation. pages 28–37. Association for Computational Linguistics, 2020.

C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

D. Miller. Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*, 2019.

R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, page 280–290, 2016.

L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez. Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach. *Knowledge-Based Systems*, 159:1–8, 2018.

C. M. Souza, M. R. Meireles, and P. E. Almeida. A comparative study of abstractive and extractive summarization techniques to label subgroups on patent dataset. *Scientometrics*, 126(1):135–156, 2021.

L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618, 2007.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

D. Xiao, H. Zhang, Y. Li, Y. Sun, H. Tian, H. Wu, and H. Wang. Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3997–4003, 2020.

J. Xu, Z. Gan, Y. Cheng, and J. Liu. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031. Association for Computational Linguistics, 2019.

J. Zhang, Y. Zhao, M. Saleh, and P. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 6197–6208. Association for Computational Linguistics, 2020.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

A Preliminary Study for Literary Rhyme Generation based on Neuronal Representation, Semantics and Shallow Parsing

Luis-Gil Moreno-Jiménez[1], **Juan-Manuel Torres-Moreno**[1], **Roseli S. Wedemann**[2]

[1]Laboratoire Informatique d'Avignon – Avignon Université, Avignon, France

[2]Inst. de Matemática e Estatística – Univ. do Estado do Rio de Janeiro, RJ Brazil

luis-gil.moreno-jimenez@univ-avignon.fr,

juan-manuel.torres-moreno@univ-avignon.fr, roseli@ime.uerj.br

Abstract. *In recent years, researchers in the area of Computational Creativity have studied the human creative process proposing different approaches to reproduce it with a formal procedure. In this paper, we introduce a model for the generation of literary rhymes in Spanish, combining structures of language and neural network models The results obtained with a manual evaluation of the texts generated by our algorithm are encouraging.*

1. Introduction

For many years, research in Artificial Intelligence (AI) has directed efforts towards automating processes to perform specific academic, industrial or economic tasks for society. However, the investigation and development of procedures for the automation of human artistic and creative processes has not had as much attention due to the complexities involved in these activities. Procedures developed for these purposes involve mathematical-computational methods designed to process and learn from a large quantity of digital data, so as to detect patterns in order to simulate the creative process (CP), as explained by Boden in [Boden 2004].

In this paper, we introduce a model for the generation of rhymes with literary components. Our proposal is based on findings detailed in [Moreno-Jiménez et al. 2020a], where Automatic Text Generation (ATG) techniques are combined with neural network (NN) based models, such as the *Word2vec* algorithm [Mikolov et al. 2013b], for the generation of literary texts. In Section 2, we present some of the literature regarding literary text generation, focusing on methods related to this paper. In Section 3, we explain the RIMAX model used to generate the rhyming words. In Section 4, we describe the corpora used for the learning phase of our models. Then, in Section 5, we explain the methodology implemented for the generation of rhymes. We show some experiments and examples in Section 6, as well as the results of evaluations conducted by humans. Finally, we present conclusions and propose possible future works in Section 7.

2. Related Work

There has been much interest and work in the area of ATG with different and interesting goals, regarding the different types of texts. Many of the proposed algorithms are based on neural networks. In [Kiddon et al. 2016], the authors generate coherent text using a recurrent neural network (RNN) and a *neural checklist model*. Their RNN predicts the best

context from a list of keywords. Another RNN approach is proposed in [Clark et al. 2018] for generating narrative text, such as fiction or news stories. Entities mentioned in the text are represented by vectors, which are updated as the text generation proceeds as they represent different contexts and guide the RNN in determining the vocabulary to be retrieved in order to generate a narrative.

We note that there are other ATG techniques, such as *text realization* that creates text in a human language, *e.g.* English or French, from a syntactic representation [Molins and Lapalme 2015]. Oliveira has written a survey of work treating the automatic generation of poetry [Oliveira 2017], and presents his own method for generating poems based on the use of templates (*canned text*) in [Oliveira and Cardoso 2015]. Another work based on canned text is presented in [Agirrezabal et al. 2013], which generates strophes of verses for Basque poetry. In [Zhang and Lapata 2014], a RNN was proposed for the generation of Chinese poetry based on learning of known text structure.

3. Semantic Rhyme

RIMAX is the first automatic system for detecting semantic rhymes in Spanish [Urrea and Torres-Moreno 2019]. It contains the following ingredients: 1) a rhyming dictionary, 2) the set of definitions of those rhymes and 3) a strategy to measure semantic proximity. This procedure can be applied to different romance languages, although we have chosen the Spanish language spoken in Mexico, given the availability of some useful resources and tools, such as the Dictionary of Mexican Spanish (DEM)[1] and the Rhyming Dictionary [Medina Urrea 2018].

Rhyming dictionaries gather words according to rhyming patterns. *Consonant* rhymes share ending sequences of vocalic and consonant sounds and *assonant* rhymes share similar vowel sounds. These two classes are thus based on pronunciation features, not on writing patterns. Also, since consonance and assonance depend on the stressed syllable, words which end with a stressed syllable are grouped together, those whose stressed syllable is the next to last appear together, and so on[2]. In this paper we have used the nomenclature of the DEM to automatically generate a phonological transcription.

3.1. Rhyme Ranking by Definition Similarity

Online dictionaries offer useful and simple advantages such as ranking and ordering of results. From a language perspective, it is interesting that text mining techniques can be applied to accomplish this. In fact, text similarity measures can be used to determine how similar word definitions are, i. e. measuring definition similarity.

Let D be a dictionary containing the set of defined words w and the set of definitions d. Since a word may have several senses, let d_{ij} be the jth definition of word w_i in D. Similarly, let d'_{kl} be the lth definition of word w_k. Also, let $\vec{v_{ij}}$ and $\vec{v'_{kl}}$ be vectors where the frequencies of lemmatized or ultrastemmized [Torres-Moreno 2012] content words of definitions d_{ij} and d'_{kl} are stored. Then, the similarity between d_{ij} and d'_{kl} can then be measured using the well-known quantity called *cosine similarity measure*, s_c.

[1] *Diccionario del español de México*, `https://dem.colmex.mx/`.

[2] For example, the penultimate syllables of the following Spanish words are the stressed syllables: *angula, chula, mula, chamula*. So these words should appear together in a rhyming dictionary.

In order to find semantic rhymes, each member of the rhyming set of word x will be weighed according to how similar its definition is to that of x, using the similarity measurement s_c. Hence, given a query word, consonance and assonance lists are generated and ordered by the calculated similarity among definitions. The program RIMAX allows us to select some parameter values for experiments.

4. Corpus

The corpus **MegaLite-Es** was used to train our model. It consists of $5\ 075$ literary documents (mainly books) in Spanish. This corpus can be useful for different NLP tasks. The documents of **MegaLite-Es** were obtained from different personal collections and, for copyrights reasons, the distribution of the original documents is not possible. Instead of this, in [Moreno-Jiménez and Torres-Moreno 2021] the authors propose some alternative resources.

4.1. Corpus Structure

The 5 075 documents in **MegaLite-Es** were written by 1 336 Spanish-speaking authors and official translations from languages other than Spanish. The documents represent different literary genres such as plays, poems, tales, essays, etc. We thus consider that this corpus is suitable for training *Word2vec* models.

The original documents, in heterogeneous formats[3] were processed to be converted into *utf8* encoded documents. A segmentation process divided the texts into sentences, corresponding to regular expressions, using a tool developed in PERL 5.0. Some undesirable data like: mutilated words, strange symbols and an unusual disposition of paragraphs could not be treated, although these situations are usual when dealing with this kind of corpora. Some characteristics of **MegaLite-Es** are detailed in Table 1.

Table 1. Characteristics of MegaLite-Es corpus (M = 10^6 and K = 10^3).

MegaLite-Es	Sentences	Tokens	Characters	Authors
Overall	15 M	212 M	1 265 M	1 328
Avg per document	3 K	41.8 K	250 K	–

MegaLite-Es has the advantage of being very extensive and suitable for automatic learning. It has, however, the disadvantage that many of its sentences consist of general language, without literary elements (stylized vocabulary or literary figures). However, these sentences often allow for fluent reading and provide the necessary links between the ideas expressed in the text, although they could imply some noisy results. As numbers identifying pages, chapters, sections or index could imply errors in the detection of sentences during segmentation, a manual process was performed to remove this undesirable data, although these errors may occur in a linguistic corpus with unstructured text.

5. Text Generation Model

In this section, we describe the model we have proposed for the generation of literary rhymes. The model consists of two steps described as follows.

[3] pdf, txt, html, doc, docx, odt, etc.

5.1. First Step: Canned Text Method

We implemented a *Canned Text* method, which has the advantage of being efficient for syntactic analysis in ATG tasks [van Deemter et al. 2005], to generate grammatical templates named Partially Empty Grammatical Structures (PGSs). Each PGS is composed of Part-of-Speech (POS) tags[4] and function words[5]. The POS tags are retrieved by Freeling [Padró and Stanilovsky 2012]. PGSs are created from a template set, called *TempSet*, that consists of sentences selected manually from the **MegaLite-Es** corpus, according to the following rules.

- Each sentence must express a concrete idea.
- Each sentence must have a length N, such that $5 \leq N \leq 10$.
- The sentence should contain at least three lexical words[6].

For generating rhymes, the process begins by selecting two original sentences $f1$ and $f2$ from *TempSet* with length N, $5 \leq N \leq 10$. Sentences $f1$ and $f2$ must satisfy two additional conditions: (*1*) both sentences must finish with a lexical word; (*2*) the lexical words finishing the sentences must have the same grammatical inflection. These sentences are analyzed with FreeLing to detect lexical words that are replaced by POS tags. We concentrate on lexical words because they provide the most meaningful information in a text [Bracewell et al. 2005]. Function words are retained in the sentence, as these are useful for maintaining the grammatical coherence and we therefore do not change them.

The idea is to generate artificial sentences from "human" sentences, respecting their grammatical structure and substituting only the lexical words by words with the same linguistic inflection but a different meaning. This technique of text generation is well-known as homo-syntax. In contrast to the paraphrase that keeps the same meaning between the original and the generated texts and changes the grammar, homo-syntax seeks to generate a new text with a different meaning than the original text, although with the same grammatical structure. In Fig. 1, we show an illustration of the proposed model. The filled boxes represent function words, whereas the empty boxes represent the lexical words that are replaced by POS tags. Once the pair of sentences has been transformed into a PGS, it will be further changed by the procedure of the second step.

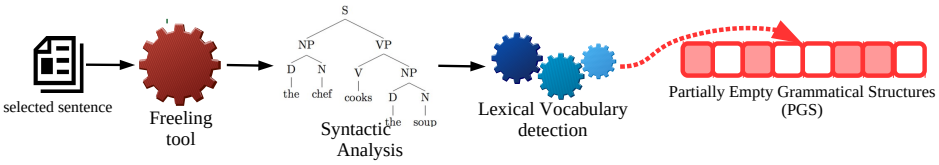


Figure 1. First step: Canned Text Method

5.2. Second step: Vocabulary selection

In this step, the POS tags in the PGSs are replaced by a vocabulary selected with the *Word2vec*[7] model. In a trained *Word2vec* model, each word, j, in the vocabulary is represented by a vector $\vec{L}_j$ with numerical valued elements. This allows for the implementation

[4] A POS tag indicates the part of speech grammatical category of a word.
[5] Prepositions, pronouns, auxiliary verbs, or conjunctions.
[6] Verbs, adjectives, nouns and adverbs
[7] *Word2vec* belongs to a group of ANN models, used to produce embeddings [Bengio et al. 2013].

of different mathematical procedures such as, for example, interpreting a semantic relation between a pair of words by calculating the cosine similarity between their vectors. These numerical vector representations are called *embeddings*.

The hyper-parameters configured for the *Word2vec* training were: *Iterations* = **10** (the number of training epochs over the **MegaLite-Es** corpus), *Minimum count* = **3** (the minimum number of times that a word must appear in the corpus to be included in the model's vocabulary), *Vector size* = **100** (the dimension of vectors, the *embeddings*) and *Window size* = **5** (the radius of adjacent words that will be related to the current word within a sentence, during the training phase of the model). We trained the model following the skip-gram procedure [Mikolov et al. 2013a]. Using the **MegaLite-Es** corpus for training, a trained model of 346 616 *Embeddings*[8] is obtained.

5.2.1. Word2vec Model

For the replacement, we used the analogical reasoning task introduced in [Mikolov et al. 2013a]. This reasoning consists of considering the relation between words, e. g. "France", "Paris", "Spain" and a missing word x. We suppose that "France", "Paris" and "Spain" are words that belong to the vocabulary of a corpus **CorpA** that was used to train *Word2vec*, and therefore, $\vec{Paris}, \vec{France}$, and $\vec{Spain}$ are the corresponding vectors associated to these words after training, respectively. The word x is then determined by finding a vector $\vec{x}$ associated to a word in **CorpA**, such that $\vec{x}$ is closest to $\vec{y}$ = $\vec{Paris} - \vec{France} + \vec{Spain}$, according to the cosine similarity between $\vec{y}$ and $\vec{x}$ (see Eq. (2)). This specific example is considered to have been answered correctly, if $\vec{x}$ is the vector corresponding to "Madrid" in the vocabulary of **CorpA**. For the replacement of POS tags, we consider the three following words, their embedding vectors and Eq. (1),

Q: the context given by the user as a single query word,
O: the original word in $f1$ or $f2$ that is replaced by the POS tag,
A: the word adjacent to O on the left, in sentence $f1$ or $f2$, if it exists,

$$\vec{y} = \vec{A} - \vec{O} + \vec{Q}\,, \tag{1}$$

where $\vec{y}$ is the vector which we will use to choose the $M = 4\ 000$ closest embeddings. We rank the M embeddings in a list $\mathscr{L}$, by calculating the cosine similarity between the j^{th} embedding, $\vec{L_j}$, and $\vec{y}$,

$$\theta_j = \cos(\vec{L_j}, \vec{y}) = \frac{\vec{L_j} \cdot \vec{y}}{||\vec{L_j}|| \cdot ||\vec{y}||} \quad 1 \leq j \leq M. \tag{2}$$

$\mathscr{L}$ is ranked according to decreasing θ_j.

If we are replacing the first POS tag, then $A = None$, so we only compute $\vec{y} = \vec{O} + \vec{Q}$. For example, for the *Query* word *love* and the sentence: *I play the guitar*, we will replace the verb *play* and the noun *guitar*. Starting by the verb *play*, we compute $\vec{y} = \vec{play} + \vec{love}$ to get the ranked list $\mathscr{L}$. Some examples of returned embeddings are: *to like, to role, enchanting* and *abandon*. This list is then used in the analysis performed by the language model based on bigrams.

[8]Term used for the numerical representation of words for NLP, typically in the form of a real-valued vector that encodes the meaning of the word.

5.2.2. Language Model Analysis (Bigrams)

This step consists of calculating the conditional probability of a word, given a preceding word, that is

$$P(w_n|w_{n-1}) = \frac{P(w_n \wedge w_{n-1})}{P(w_{n-1})}\,. \tag{3}$$

Each bigram in the **MegaLite-Es** corpus has been detected and computed in this way. As a result, we have a new list of bigrams, LB. The bigrams are composed only by lexical and function words, ignoring punctuation, numbers and symbols. For each element in $\mathscr{L}$, we configure two bigrams, as b_1 and b_2, where:

- b_1 is the adjacent word to the left of O in $f1$ or $f2$, concatenated with the current analyzed word, L_j, and
- b_2 is the current analyzed word L_j concatenated with the adjacent word to the right of O, in $f1$ or $f2$.

We calculate the arithmetic mean, bm, of the frequencies of occurrence of b_1 and b_2 in LB. If O is the first (last) word in the sentence, we do not calculate any mean, and bm will be simply equal to the frequency of b_1 (b_2). The process is repeated for the M elements in $\mathscr{L}$. The values of bm are combined with the cosine similarities for each L_j, to re-rank $\mathscr{L}$ as

$$\theta_j = \frac{\theta_j + bm_j}{2}\,, \quad 1 \leq j \leq M\,. \tag{4}$$

Finally, we take the word at the top of the list as the chosen candidate to substitute O. The idea is to select the word that is semantically most similar to $\vec{y}$, based on the analysis accomplished with *Word2vec*, and maintain coherence of the text obtained with guidance of the language model. The process is repeated for each word in $f1$ and $f2$, except when we replace the last word in the second sentence, $w2_L$. To replace $w2_L$, we present the word that substituted $w1_L$ in $f1$ as input to RIMAX. RIMAX returns a ranked list LR with consonant and assonant rhymes related to $w1_L$. A score, LR_w is attributed to each word, w in LR, corresponding to a semantic similarity measure, which results from the semantic-phonetic analysis performed by a hybrid, automatic and manual process. The scores are normalized in the interval $[0-1]$. The scores in LR are combined with the scores in $\mathscr{L}$. For this, an average score is calculated for each pair of elements LR_w and L_w, where L_w corresponds to the element of $\mathscr{L}$ with the information referring to w.

$$\theta_w = \frac{\theta_w + LR_w}{2}\,, \qquad \forall w \in LR\,. \tag{5}$$

The words in LR that do not exist in $\mathscr{L}$ are also considered, and since LR_j is divided by 2, this strategy allows us to prioritize the elements contained in both lists. The new values in $\mathscr{L}$ are then processed with the language model as already described. Then we take the element in $\mathscr{L}$ in the first place, which contains the best semantic and coherent rhyme. Finally, a morphological analysis is performed once again with FreeLing, in order to convert the selected word into the correct inflection specified by the POS tag, if necessary. For that, we carry out conjugations and genre or number conversions.

The result is a newly generated pair of phrases that does not exist in the **MegaLite-Es** corpus, where $w2_L$ must rhyme with $w1_L$. The model is illustrated in Fig. 2, where

the two PGSs (for $f1$ and $f2$) can be appreciated at the top of the illustration. Both structures are sending inputs to the *Word2vec* model, which receives Q, A and O in order to generate the list $\mathscr{L}$ with the vocabulary related to the inputs. It can be observed that the RIMAX module outputs its result to $f2$. RIMAX receives $w1_L$ and generates the list of rhymes LR, and then $\mathscr{L}$ and LR are combined by Eq. (5) to update the $\mathscr{L}$ list. The list $\mathscr{L}$ with the vocabulary is then sent to the language model module, for the selection of the most coherent option. Finally, the best option is processed with FreeLing, in order to make it respect the grammatical information provided by the POS tag in the PGS (to preserve inflection). In the Fig. 2, we have marked in blue the **First step** where the semantic vocabulary list is generated to subsequently be sent to the **Second step**, the Language Model module marked in green. The pink section, the **Rhyme analysis**, is an independent section that is only executed once in the process.

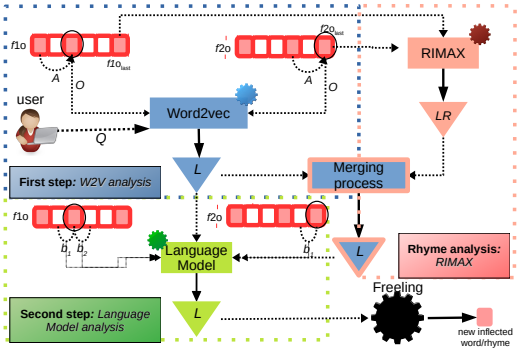


Figure 2. Second Step: Vocabulary Selection

We can illustrate an example in Spanish of our model as follows:

1. **Templates generation** (Canned Text): *Jamás he sido más [AQMS]; yo era ya un [NCMS]. / [VMI3S] el [NCMS] rápidamente hacia las [NCFP] que [VMII3P] el [NCMS].*
2. **Vocabulary selection** (Word2vec): *Jamás he sido más [afectuoso]; yo era ya un [NCMS]. / [Corría] el [sol] rápidamente hacia las [cumbres] que [anteponían] el [NCMS].*
3. **Rhymes selection** (RIMAX): *Jamás he sido más [afectuoso]; yo era ya un [**ofrecimiento**]. / [Corría] el [sol] rápidamente hacia las [cumbres] que [anteponían] el [**firmamento**].*

6. Experiments and Evaluation

In preliminary tests of our proposal, we generated and evaluated 44 pairs of rhyming sentences that were generated using PGSs created from sentences in the **MegaLite-Es** corpus. The PGSs respect the rules specified in Section 5.1. The new contexts are given by eleven different queries: *amor, odio, tristeza, alegria, sol, luna, hombre, mujer, bosque, desierto, mar* (love, hate, sadness, joy, sun, moon, man, woman, forest, desert, sea). An examples of the generated sentences are listed below, where we show the queries, the sentences in Spanish in **bold** print, and in *italic* their translation.

sadness: **El sol de mediodía encapota sobre la inexpresable niebla de mi bosque.** | *The midday sun overlays the inexpressible mist of my forest.*
Subía el sol rápidamente hacia las desolaciones que limitaban el zopilote. | *The sun was rising rapidly towards the desolations that bordered the vulture.*

Although general ATG tasks have been widely addressed by the research community, using different automatic evaluation protocols, it is difficult to implement automatic evaluation in the case of *literary text* due to the ambiguity and subjectivity involved in its interpretation and evaluation [Boden 2004]. For this reason, we have performed a manual evaluation of our experiments, asking 6 people with a graduate degree in literature to evaluate the rhymes generated by our algorithm and their semantic relations. In previous general ATG models [Moreno-Jiménez et al. 2020b, Moreno-Jiménez et al. 2020a], criteria such as coherence and grammatical composition were evaluated. Here, we asked the evaluators to indicate if they perceived a rhyme between the last words of each sentence in a pair and also to specify their perception of the semantic relation between the two rhyming words, which could be one of the following: *any relation*, *low relation*, *acceptable relation*, *good relation* and *strong relation*. We calculated the mode and median of the evaluator's feedback, obtaining a *low relation* between the two rhyming words. This was to some extent expected because, when the model looks for the second rhyming word, the semantic analysis is performed considering not only the word to rhyme, but also the general context (the *query*) and the adjacent word. For this reason, it is expected that, in some cases, the semantic relation between the two rhyming words cannot be preserved, although some relation was always perceived. For the evaluation of rhyme, we obtained encouraging results with a perception of rhymes in 61% of the pairs of sentences.

7. Conclusions and Future Work

We have proposed a model capable of generating rhyming sentences in Spanish, although with a weak semantic relation between them. This can be improved by altering the semantic analysis, when selecting the second rhyme. We have presented preliminary results showing that the model generates literary sentences that integrate semantic aspects with rhymes. Nevertheless, it must be considered that this task is still an open problem and further models, their extensions and generalizations, and experiments may confirm or improve the results that we have obtained. We expect to perform more complex and extensive evaluations, and analyse more criteria, such as coherence, by generating more sentences and recruiting more evaluators. We also plan to conduct experiments in other languages, like French or Portuguese.

Acknowledgment: We thanks CONACYT (Mexico), grant number 661101 and Agorantic (Avigon Université, France) by their financial support.

References

Agirrezabal, M., Arrieta, B., Astigarraga, A., and Hulden, M. (2013). Pos-tag based poetry generation with WordNet. In *European Workshop on NLG '13*, pages 162–166, Sofia, Bulgaria. ACL.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Boden, M. A. (2004). *The creative mind: Myths and Mechanisms*. Routledge.

Bracewell, D., Ren, F., and Kuriowa, S. (2005). Multilingual single document keyword extraction for information retrieval. In *Proc. ICNLPK' 05*, pages 517–522, Wuhan, China. IEEE.

Clark, E., Ji, Y., and Smith, N. A. (2018). Neural text generation in stories using entity representations as context. In *Proc. NACACL-HLT '18*, volume 1, pages 2250–2260, New Orleans, Louisiana.

Kiddon, C., Zettlemoyer, L., and Choi, Y. (2016). Globally coherent text generation with neural checklist models. In *Proc. EMNLP '16*, pages 329–339, Austin, Texas. Association for Computational Linguistics.

Medina Urrea, A. (2018). *Diccionario de rimas asonantes y consonantes del español de México*. El Colegio de México, Mexico.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, *ICLR '13*, Scottsdale, Arizona, USA. ICLR.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *NACACL-HLT '13*, pages 746–751, Atlanta, USA.

Molins, P. and Lapalme, G. (2015). JSrealB: A bilingual text realizer for web programming. In *Proc. ENLG '15*, pages 109–111, Brighton, UK. ACL.

Moreno-Jiménez, L.-G., Torres-Moreno, J.-M., and Wedemann, R. S. (2020a). Literary natural language generation with psychological traits. In *Proc. NLPIS '20, LNCS*, volume 12089, pages 193–204, Cham. Springer.

Moreno-Jiménez, L.-G., Torres-Moreno, J.-M., Wedemann, R. S., and SanJuan, E. (2020b). Generación automática de frases literarias. *Linguamática*, 12(1):15–30.

Moreno-Jiménez, L.-G. and Torres-Moreno, J.-M. (2021). Megalite: A New Spanish Literature Corpus for NLP Tasks. In David C. Wyld, D. N. E., editor, *Proc. AIAP '21*, Zurich, Switzerland.

Oliveira, H. G. (2017). A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proc. ICNLG '17*, pages 11–20.

Oliveira, H. G. and Cardoso, A. (2015). Poetry generation with PoeTryMe. In *Proc. CCR-TCM '15*, volume 7, Paris. Atlantis Thinking Machines.

Padró, L. and Stanilovsky, E. (2012). FreeLing 3.0: Towards wider multilinguality. In *Proc. of the 8th on LREC '12*, pages 2473–2479, Istanbul, Turkey.

Torres-Moreno, J.-M. (2012). Beyond stemming and lemmatization: Ultra-stemming to improve automatic text summarization. *ArXiv*, abs/1209.3126.

Urrea, A. M. and Torres-Moreno, J.-M. (2019). RIMAX: ranking semantic rhymes by calculating definition similarity. *ArXiv*, abs/1912.09558.

van Deemter, K., Theune, M., and Krahmer, E. (2005). Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24.

Zhang, X. and Lapata, M. (2014). Chinese poetry generation with recurrent neural networks. In *Proc. EMNLP '14*, pages 670–680, Doha, Qatar. ACL.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Structural Characterization and Graph-based Detection of Fake News in Portuguese

Roney Lira de Sales Santos[1], **Thiago Alexandre Salgueiro Pardo**[1]

[1]Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematical and Computer Sciences, University of São Paulo
São Carlos, Brazil

roneysantos@usp.br, taspardo@icmc.usp.br

Abstract. *The production of fake news is a problem nowadays. With social networks, fake news spread in easier and less costly ways, having the power to reach a large number of people in a short time. In this paper, we investigate graph-based approaches for fake news characterization and detection, taking into account widely used measures of graphs and complex networks. Our results show that some network measures are useful for structurally characterizing fake and true news and that machine learning-based solutions over this kind of feature produce promising results.*

Resumo. *A produção de notícias falsas é um problema dos dias atuais. Com as redes sociais, as notícias falsas se espalham de forma mais fácil e barata, podendo chegar a um grande número de pessoas em um curto espaço de tempo. Neste artigo, investigamos abordagens baseadas em grafos para caracterização e detecção de notícias falsas, levando em consideração medidas amplamente utilizadas de grafos e redes complexas. Nossos resultados mostram que algumas medidas de rede são úteis para caracterizar estruturalmente notícias falsas e verdadeiras e que soluções baseadas em aprendizado de máquina sobre esse tipo de atributo produzem resultados promissores.*

1. Introduction

Nowadays, fake news detection has become an important research topic, and the need to always evaluate the veracity of digital content has been raised by the constant spread of fake news [Figueira and Oliveira 2017]. According to [Rubin et al. 2015], in the Natural Language Processing (NLP) area, two main approaches may be used for automatically identifying fake news: linguistic and network approaches.

In linguistic approaches, there is an attempt to find linguistic clues that can differentiate fake news from true news, and these clues generally work well when analyzing text features such as lexical, morpho-syntactic, syntactic, semantic, psycho-linguistic [Pérez-Rosas and Mihalcea 2015, Pérez-Rosas et al. 2017] and readability measures [Santos et al. 2020]. In the case of network approaches, there are fact-checking models, which aim at relating the elements of an statement (e.g., by generic and domain-specific relations as "is a", "member of" and "is married to") to subsidize content verification [Ciampaglia et al. 2015]. In a related research line, graph-based approaches to detect fake news are present in the literature with considerable results, using knowledge

graphs [Ciampaglia et al. 2015, Pan et al. 2018], biclique identification, graph-based feature vector learning and label spreading [Gangireddy et al. 2020] and graph-based modeling of online communities [Chandra et al. 2020], among others. This type of approach is important to detect false information, as it allows examining beyond linguistic characteristics of the writing and determining the structural behavior of fake news.

This paper has as contribution the investigation of two graph-based approaches to deal with fake news for the Portuguese language: the SentiElection, an ensemble of global centrality measures over reference graphs, proposed by [Vilarinho and Ruiz 2018] for the task of sentiment analysis; and a complex network-based approach, in which the news are converted into complex networks and several of the most used network measures in the literature are applied, to then classify the news as fake or true. To the best of our knowledge, this is the first time that such methods are explored for fake news detection.

Trying to uncover the structural characteristics of fake and true news in Portuguese, we show that some complex network measures are interesting to distinguish them, as clustering coefficient, eigenvector, katz and pagerank values. In machine learning-based detection approaches, the network measures are used as features and some of them produce promising results, as katz and density, with K-Means and SVM techniques producing the best results. The adaptation of SentiElection to the task in hands shows that the accuracy of fake news detection increases as the reference graphs grow, although its results do not overcome the ones produced by the complex network measures.

The remainder of this paper is organized as follows. Section 2 briefly introduces the main related work in the area. Section 3 details the graph-based approaches that we explore in this paper. Experiments and results are presented in Section 4 and some final remarks are made in Section 5.

2. Related Work

Graph-based (or network-based) approaches generally address and try to solve the task of automatic fact-checking. [Thorne and Vlachos 2018] formally describe fact-checking as the task of evaluating whether the statements made, in both written and spoken language, are true, which is usually done by trained professionals. For automatic checking, the authors suggest that the input to fact-checking approaches should be triples of the form (subject, predicate, object), which facilitate fact-checking in structured (and semi-structured) databases. Such a triple is one of the main patterns we have today for the construction of graphs, where the subject and object are nodes of the graph and the predicate is the relationship (edge) between the nodes.

Approaches using knowledge graphs, with a triple being the main element, can provide a rich set of structured information related to world knowledge stored in a machine-readable format that supports the task of checking facts, being widely used in the literature [Ciampaglia et al. 2015, Shi and Weninger 2016, Pan et al. 2018], including the case of Portuguese language [Santos and Pardo 2020], which had 74% average accuracy when simple statements were checked (i.e., *Brasília is the capital of Brazil*).

Complex networks are often used in detecting the spread of fake news [Lind et al. 2007, Alassad et al. 2019]. In [Paluch et al. 2018], the authors tried to find a method that executes the task of identification of the propagation source in reasonable

time on large complex networks and delivers high quality localization results at the same time. [Zhou and Zafarani 2019] work with a pattern called Denser-Network to identify fake news spreaders, since the authors hypothesize that fake and true news spreaders show different graph properties, reaching 90% accuracy in this approach. However, to the best of our knowledge, there are no works that use measures extracted from a complex network to directly characterize and classify news text as fake or true.

In the next section we introduce the techniques that we explore in this paper.

3. Graph and Complex Network-based Approaches

3.1. SentiElection Approach (SEA)

The SentiElection approach is the method proposed by [Vilarinho and Ruiz 2018], which is a voting system over three measures (katz, eingenvector and pagerank) computed for a text in relation to reference graphs. A text of interest is classified according to the reference graph that it best fits (as indicated by the taken measures in the voting strategy).

In this work, SEA behaves as a classifier in detecting fake news. The approach is made with two graphs formed by news words: one graph with fake news (G-FAKE) and other with true news (G-TRUE). The graphs are formed with a variation of the number of edges that each word has, called the word frame[1].

True and fake news from the Fake.Br Corpus[2] [Monteiro et al. 2018, Silva et al. 2020] are used as input to the classifier. Of the 3,600 fake news items present in the corpus, a certain number of news items were randomly selected to form the fake news graph. The same process was used for the construction of the true news graph. Details of the experiments are shown in Section 4.1.

3.2. Complex Network Measures-based Classifier (CNMC)

The Complex Network Measures-based Classifier is a classifier that works with measures widely used in Complex Networks approaches, namely: betweenness, closeness, eingenvector, katz, pagerank, hubs and authorities, cluster coefficient (and its average), correlation, transitivity and density. The measures are briefly explained[3] in Table 1.

The mentioned measures were applied in all news from Fake.Br Corpus. Firstly, the graph of each news item was generated, following the same procedures that were used to generate the graphs of the SEA, but with one change: the word frame is 1, i.e., each word is linked only to its adjacent words.

With the generated graphs, the second part of the process was to compute the complex network measures. Such measures are extracted from the implementations in the NetworkX framework [Hagberg et al. 2008], using the default implementations of each measure, without additional configurations, except for the centrality measures katz, pagerank and hubs and authorities, where a setting of maximum interaction of the algorithm is

[1]The word frame represents the distance between the analyzed word and the words in sequence. If the distance is 1, the edge in the graph would be from word n to n+1; if the distance is 2, the edge in the graph would be from word n to n+1 and to n+2, where n is the position of the word in the sentence.

[2]Details about the dataset, such as how and when the data was collected, the sources and news annotation, can be found in [Monteiro et al. 2018] and [Silva et al. 2020].

[3]According to NetworkX documentation available at https://networkx.org/. Further details about the measures can be found in [Morais and Prati 2013] and [Comin et al. 2020].

Type	Metric	Explanations
Centrality	Betweenness	Quantifies the participation of a node v in paths of minimum length
	Closeness	Measures the proximity of node v to all nodes in the network
	Eingenvector	Brings out the centrality of a node v based on the centrality of their neighbours
	Katz centrality	Computes the centrality for a node based on the centrality of its neighbors, on local and global influence
	Pagerank centrality	Computes a ranking of the nodes in the graph G based on the structure of the incoming links
	Hubs	Estimates the node value based on outgoing links
	Authorities	Estimates the node value based on incoming links
Clustering	Cluster coefficient and Cluster coefficient average	Quantifies the fraction of possible triangles through that node that exist. A triangle is a set of three nodes, where each node has a relationship to all other nodes
	Correlation	Computes the similarity of connections in the graph with respect to the node degree
	Transitivity	Computes the fraction of all possible number of ”triads” (two edges with a shared vertex)
Other	Density	Measures of how is a graph in which the number of edges is close or far to the max number of edges

Table 1. Complex Network measures

determined (1,000 epochs). This configuration was necessary because interactions below this value did not result in reliable results.

Besides attempting to structurally characterizing the news, the measures served as input features to various machine learning techniques, both symbolic and statistical. The analysis were divided into verifying the results with the combined features and with individual features. Details of the cited experiments are reported in Section 4.2.

4. Experiments and Results

To test the approaches reported in this paper, some experiments were performed. In all the experiments, the truncated versions of the fake and true news in the Fake.Br Corpus were used (in order to guarantee similar text size and more reliable experiment results).

4.1. The SEA

For the analyses of the SEA experiments, graphs of different sizes were built to observe the accuracy of the classifier when news are analyzed by the amount of information present. Graphs were formed with 100, 200, 300, 400, 500, 600, 700 and 800 news, both fake and true, totaling 16 graphs with all news being randomly selected for each set. The test was done with 1,000 news, also randomly selected, being 500 fake news and 500 true news. It was necessary to keep such balance for more faithful results.

The pipeline used in the experiment had three steps: 1) the extraction of news tokens; 2) the construction of the graph[4]; and 3) the analysis of the SEA classification results. From the selected news set, each news item is tokenized with the NLTK tokenizer [Bird et al. 2009] and is tagged by NLPNet [Fonseca et al. 2015]. If the token is an adverb, adjective, noun (common or proper) or a verb, the token is selected to form the graph; if the token is a stopword, it is discarded. Using only content words ensures more informative links between words. The words are then linked to form the graph, using word frame = 3, since [Vilarinho and Ruiz 2018] demonstrated that this was the best configuration in their experiments. The graphs were formed using the NetworkX framework. Finally, for the test, each graph formed by the union of the *reference graph* + *news graph* is submitted to the centrality calculation of the eigenvector, katz and pagerank measures, which work as a voting system, indicating the category of the news (true or fake) according to the graph that produces the best results.

Analyzing the 1,000 news in the 16 G-FAKE and G-TRUE graphs, we synthesize the results in Figure 1. The evaluation measures were the geometric mean (as in [Vilarinho and Ruiz 2018] - blue line) and accuracy (orange line).

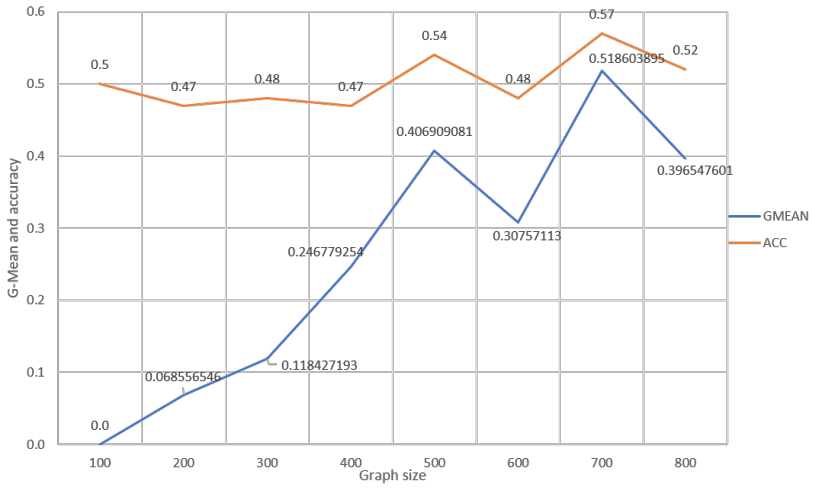


Figure 1. Results of the SEA experiment

The performance of the algorithm increases proportionally to the size of the test graph, with an outlier for the graph of size 600, understandable by the randomness of the news selection process. The best performance was when the graph reached size 700, with a drop in performance when it reached size 800.

4.2. The CNMC

Unlike the SEA, the news graphs for the CNMC experiment were unique, i.e., a single graph for each news was formed for the analysis. The selected word frame was 1, with each word being linked with its immediate neighbors.

[4]Details about graph construction can be found in [Vilarinho and Ruiz 2018].

Experiments were carried out with graphs with and without stopwords[5]. It is noteworthy that the stopwords were extracted from the NLTK, similar to what we use in the SEA, however, here we use the full return of NLTK stopwords, without distinguishing part-of-speech classes. For each of the types of graphs (fake or true), the 12 measures of complex networks cited before were extracted by the NetworkX framework.

It was possible to make a first comparative analysis between the complex network measures from fake and true news. Table 2 shows the comparison between the individual metrics in each of the news types and graph composition (with or without stopwords). The numbers represented in Table 2 take into account when **metric(true)** $>$ **metric(fake)** happened and the corresponding percentage. It is important to mention that 3,600 comparisons were made between news, since Fake.Br Corpus is an aligned dataset, with each fake news having a corresponding true one.

Measure	**with stopwords**		**without stopwords**	
	metric(true)>metric(fake)	%	metric(true)>metric(fake)	%
Betweenness	1,756	0.48	1,592	0.44
Closeness	1,664	0.46	1,446	0.40
Eigenvector	1,645	0.45	**1,379**	**0.38**
Katz	1,660	0.46	**1,400**	**0.38**
Page Rank	1,705	0.47	**1,362**	**0.37**
Hubs	1,836	0.51	1,882	0.52
Authorities	1,751	0.48	1,685	0.46
Cluster Coeff Avg	1,758	0.48	1,444	0.40
Cluster Coeff	**585**	**0.16**	**837**	**0.23**
Correlation	1,630	0.45	1,957	0.54
Transitivity	1,549	0.43	1,443	0.40
Density	1,828	0,51	1,673	0.46

Table 2. Comparison based in absolute values of the metrics

It is possible to see that the biggest difference occurred in the cluster coefficient, where only 16% of the true news with stopwords and 23% of the true news without stopwords had this measure with a value greater than the fake news. The other measures reached very close values, being the difference a little more relevant in the centrality measures eigenvector, katz and pagerank when the graphs did not contain stopwords.

The next experiment was to use the measures extracted from the graphs of each news to create classifiers. Nine Machine Learning (ML) techniques were selected: Decision Tree (DT), Naive Bayes (NB), Random Forest (RF), K-Nearest Neighbors (KNN), Support Vectors Machine (SVM), Multi-Layer Perceptron (MLP), OneRule (OneR), One Class (OC) and K-Means (KM). The implementations followed the pattern of Scikit-Learn framework [Pedregosa et al. 2011], except for the OneRule classifier, which implementations of the MLxtend library [Raschka 2018] were used. Standard implementations of Scikit-Learn and MLxtend were used in all approaches except for: MLP, where training

[5]Stopwords may be a sensible issue in complex network approaches, as they may significantly change the network topology and the corresponding measures.

was limited to 1,000 training epochs; OneClass, where training used the kernel coefficient = auto, i.e., using $1/n$ features; and K-Means, where the number of clusters was 2.

Two sets of experiments were performed: (i) classifiers with individual features and (ii) classifiers with combined features. In (i), each classifier received as input the individual values of each metric of each news item. For example, an SVM model was created that received as input the values of the betweenness metric for each news item. The model would then draw a cutting line where values above or below it would be fake or true. This experiment was interesting to verify if some metric by itself could be discriminating in news classification, based in what we found in the Table 2. In (ii), all news features were combined as input to one of the machine learning models.

Table 3 contains the experiments with the best results for each measure in (i). Each classifier received as input the individual measures of the news with and without stopwords. Cross-validation with $k = 5$ was the evaluation method and the accuracy results show the average of the executions of the ML approaches (10 executions).

Measure	**with stopwords**		**without stopwords**	
	model(s)	acc	model(s)	acc
Betweenness	DT, RF, SVM, MLP	0.51	NB, SVM, MLP	0.54
Closeness	NB, SVM, MLP, KM	0.52	NB, MLP	0.56
Eigenvector	NB, SVM, KM	0.53	MLP	0.56
Katz	**KM**	**0.63**	SVM, MLP	0.55
Pagerank	DT, RF, SVM	0.52	SVM, MLP	0.56
Hubs	OC	0.58	DT, RF	0.53
Authorities	**KM**	**0.67**	SVM	0.53
Cluster Coeff Avg	NB	0.51	DT, RF	0.53
Cluster Coeff	DT, KNN, MLP	0.51	DT, KNN	0.53
Correlation	NB, SVM, MLP	0.53	SVM, MLP, KM	0.53
Transitivity	KM	0.61	KM	0.53
Density	**KM**	**0.69**	DT, RF	0.52

Table 3. Results of the experiments with individual measures on the ML approaches

The best results come from graphs with stopwords. The density measure reached the best accuracy, with 69%, followed by authorities with 67% of accuracy and the katz measure with 63%. It is also interesting to note that the best results came from the K-Means approach. The graphs without stopwords had results close to random (≈50%), with the best accuracies coming from closeness and pagerank, reaching 56%.

Experiment (ii) took into account all measures as input features to the ML approaches. The results achieved are presented in Table 4. Again, the evaluation method used was the cross-validation with $k = 5$ and the accuracy results show the average of the executions of the MLs approaches (10 executions).

SVM obtained the best accuracy in this experiment, reaching 60% with graphs without stopwords, followed by MLP with 59%, also with the same setup. K-Means, which was the ML approach that had the best results when using the individual features, reached only 54% accuracy, below what was achieved in the previous experiment.

ML approach	with stopwords	without stopwords
Decision Tree	0.51	0.55
Naive Bayes	0.54	0.58
Randon Forest	0.54	0.58
KNN	0.53	0.57
SVM	0.55	**0.60**
MLP	0.55	0.59
OneR	0.50	0.51
One Class	0.24	0.24
K-Means	0.48	0.54

Table 4. Results of the experiments with combined measures on the ML approaches

Overall, the achieved results are encouraging, showing that true and fake news do have distinguishing structural characteristics. The use of these characteristics may support the identification of fake content in different ways, possibly helping overcoming the limitations of some linguistic-based approaches that check the veracity of news by evaluating their writing style and linguistic choices. Bringing additional structural knowledge to the situation may allow to deal with more difficult cases, for instance, statements with half-truths

The use of complex network measures has not been much explored for the detection of fake news. This paper was a first attempt in this endeavour. The addition of new measures present in the literature may be important to increase the accuracy of the classification and remains for future work, as we comment in the following section.

5. Final Remarks

This work focused on structural approaches for characterizing and detecting fake news for Portuguese. We adapted the SentiElection method [Vilarinho and Ruiz 2018] and explored measures of complex networks, showing that fake and true news do have distinguishing structural characteristics. The classification results did not overcome the ones achieved in linguistic-based approaches (see, e.g., [Silva et al. 2020]), but we believe that they may help improving such approaches.

Future work includes performing deeper analyses of network measures, looking for other structural patterns, and tackling more challenging phenomena, as half-truth and post-truth statements. It may also be interesting to perform cross-lingual tests to check whether the learned patterns keep the same across different natural languages.

The interested reader may find more information about this work at the web portal of the POeTiSA project[6].

Acknowledgments

The authors are grateful to CAPES, USP Research Office (PRP #668), and the Center for Artificial Intelligence (C4AI) of the University of São Paulo, sponsored by IBM and FAPESP (grant #2019/07665-4).

[6] https://sites.google.com/icmc.usp.br/poetisa

References

Alassad, M., Hussain, M. N., and Agarwal, N. (2019). Finding fake news key spreaders in complex social networks by using bi-level decomposition optimization method. In Agarwal, N., Sakalauskas, L., and Weber, G.-W., editors, *Modeling and Simulation of Social-Behavioral Phenomena in Creative Societies*, pages 41–54.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition.

Chandra, S., Mishra, P., Yannakoudakis, H., Nimishakavi, M., Saeidi, M., and Shutova, E. (2020). Graph-based modeling of online communities for fake news detection.

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., and Flammini, A. (2015). Computational fact checking from knowledge networks. *PloS one*, 10(6):e0128193.

Comin, C. H., Peron, T., Silva, F. N., Amancio, D. R., Rodrigues, F. A., and da F. Costa, L. (2020). Complex systems: Features, similarity and connectivity. *Physics Reports*, 861:1–41.

Figueira, A. and Oliveira, L. (2017). The current state of fake news: challenges and opportunities. *Procedia Computer Science*, 121:817–825.

Fonseca, E. R., Rosa, J. a. L. G., and Aluísio, S. M. (2015). Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. *Journal of the Brazilian Computer Society*, 21(1):2.

Gangireddy, S. C. R., P, D., Long, C., and Chakraborty, T. (2020). Unsupervised fake news detection: A graph-based approach. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, page 75–83.

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, pages 11 – 15.

Lind, P. G., da Silva, L. R., Andrade, J. S., and Herrmann, H. J. (2007). Spreading gossip in social networks. *Phys. Rev. E*, 76:036117.

Monteiro, R. A., Santos, R. L. S., Pardo, T. A. S., de Almeida, T. A., Ruiz, E. E. S., and Vale, O. A. (2018). Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In *Computational Processing of the Portuguese Language*, pages 324–334.

Morais, G. and Prati, R. C. (2013). Complex network measures for data set characterization. In *2013 Brazilian Conference on Intelligent Systems*, pages 12–18.

Paluch, R., Lu, X., Suchecki, K., Szymański, B. K., and Hołyst, J. A. (2018). Fast and accurate detection of spread source in large complex networks. *Scientific reports*, 8(1):1–10.

Pan, J. Z., Pavlova, S., Li, C., Li, N., Li, Y., and Liu, J. (2018). Content based fake news detection using knowledge graphs. In *The Semantic Web – ISWC 2018*, pages 669–683.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau,

D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., and Mihalcea, R. (2017). Automatic detection of fake news. *CoRR*, abs/1708.07104.

Pérez-Rosas, V. and Mihalcea, R. (2015). Experiments in open domain deception detection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1120–1125.

Raschka, S. (2018). Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack. *The Journal of Open Source Software*, 3(24).

Rubin, V. L., Conroy, N. J., and Chen, Y. (2015). Towards news verification: Deception detection methods for news discourse. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS48) Symposium on Rapid Screening Technologies, Deception Detection and Credibility Assessment Symposium*, pages 5–8.

Santos, R., Pedro, G., Leal, S., Vale, O., Pardo, T., Bontcheva, K., and Scarton, C. (2020). Measuring the impact of readability features in fake news detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1404–1413.

Santos, R. L. d. S. and Pardo, T. A. S. (2020). Fact-checking for portuguese: Knowledge graph and google search-based methods. In *Computational Processing of the Portuguese Language*, pages 195–205.

Shi, B. and Weninger, T. (2016). Fact checking in heterogeneous information networks. In *Proceedings of the 25th International Conference Companion on World Wide Web*, page 101–102.

Silva, R. M., Santos, R. L., Almeida, T. A., and Pardo, T. A. (2020). Towards automatically filtering fake news in portuguese. *Expert Systems with Applications*, 146:113–199.

Thorne, J. and Vlachos, A. (2018). Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359.

Vilarinho, G. and Ruiz, E. (2018). Global centrality measures in word graphs for twitter sentiment analysis. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 55–60.

Zhou, X. and Zafarani, R. (2019). Network-based fake news detection: A pattern-driven approach. *SIGKDD Explor. Newsl.*, 21:48–60.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

ReVera Framework: Um Framwork para rastreabilidade em fact-checking automático

João Victor de Souza[1], **Elias Cyrino de Assis**[1],
Fabrício Martins Mendonça[2], **Jairo Francisco de Souza**[2]

[1]LApIC Research Group – Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora (UFJF)
36036-900 – Juiz de Fora – MG – Brasil

[2]Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora – Juiz de Fora/MG – Brasil

{joao.souza, elias.cyrino, fabricio.mendonca, jairo.souza}@ice.ufjf.br

Abstract. *Professional fact-checking tends to be costly and scalability challenging. For this reason, a series of methods to automate this process has been emerging. However, these methods have been created with monolithic architectures, not making use of pre-built parts, and are harder to be understood by others. This work aims to propose a framework, where the development of methods can be done in a modular manner, creating workflows based on key steps that can be linked together to generate the input classification. The framework makes use of a proposed data traceability ontology to map all generated data during execution under a unified vocabulary, which facilitates communication between these independent components.*

Resumo. *A verificação manual de fatos tende a ser cara e a escalabilidade desafiadora, incentivando a busca por métodos automáticos. Porém, esses métodos foram criados com arquiteturas monolíticas, não fazendo uso de peças pré-construídas e são mais difíceis de serem compreendidos por terceiros. Este trabalho tem como objetivo propor um framework, onde o desenvolvimento de métodos pode ser feito de forma modular, criando fluxos de trabalho baseados em etapas-chave que podem ser interligadas para gerar a classificação de entrada. O framework faz uso de uma ontologia de rastreabilidade proposta para mapear todos os dados gerados durante a execução sob um vocabulário unificado, o que facilita a comunicação entre esses componentes independentes.*

1. Introdução

O consumo de informações tem sido significativamente alterado pelo notável crescimento das redes sociais, que vem se tornado bastante comum no cotidiano das pessoas [Shu et al. 2017]. A sua utilização como fonte de notícias e informação tem sido cada vez mais comum [Conroy et al. 2015]. Entretanto, nem todo conteúdo publicado é verídico [Souza et al. 2020].

No jornalismo, a verificação de fatos (do inglês, *fact-checking*), pode ser definida como uma tarefa que visa determinar a veracidade de uma informação com base em fontes externas confiáveis. Entretanto, este tem sido um problema cada vez maior devido à

quantidade de informações que os usuários precisam lidar [Shu et al. 2017], e isso acaba aumentando não apenas a demanda por verificação, mas também torna o processo cada vez mais custoso de ser realizado [Hassan et al. 2015]. Além disso, a velocidade com que as informações trafegam em redes sociais ou serviços de mensagens cria um desafio cada vez maior [Hassan et al. 2015].

A diferença entre o momento que uma informação é vista e compartilhada pelos usuários e as primeiras verificações pode ser longa demais para evitar os impactos negativos dessa disseminação. Esse efeito tem levado a uma busca por novas formas que possam diminuir cada vez mais o tempo necessário para esse processo de verificação [Shu et al. 2017]. Diversos métodos automáticos tem sido apresentados para tentar solucionar o problema. Porém, sistemas desse tipo tendem a ter um desenvolvimento complexo, devido a quantidade de fases e dados necessários para realizar a verificação [da Silva et al. 2020], fornecendo um resultado transparente ao usuário [Kotonya and Toni 2020].

Assim sendo, este trabalho visa fornecer um *framework* baseado em uma ontologia para a criação de métodos para *fact-checking end-to-end*, que são as abordagens que tratam todos os aspectos relacionados à verificação, com o objetivo de fornecer interoperabilidade entre componentes e dados de proveniência sobre o *pipeline* de processamento. As principais contribuições desse trabalho são: (1) mostrar uma ontologia para o processo de *fact-checking* automático que auxilia na reprodutibilidade e rastreabilidade; (3) apresentar um framework para a implementação de métodos de fact-checking automático com base em componentes reutilizáveis.

2. Trabalhos Relacionados

Na literatura, vários trabalhos lidam diretamente com o problema da verificação de fatos [Santos and Pardo 2020, Miranda et al. 2019, Gerber et al. 2015]. No entanto, de acordo com [Graves 2018], é importante automatizar não apenas a verificação, mas também etapas de identificação e geração de resultados. Além de verificar as declarações, também é importante monitorar fontes que possam gerar fatos que valham a pena validar, além de oferecer meios para que seu resultado seja divulgado e alcance as pessoas.

Nessa área, sistemas *end-to-end* são aqueles que implementam todos esses componentes do processo, tratando não apenas a coleta de evidências e avaliação dos dados, mas também aplicando formas de divulgação de resultados e monitoramento de fontes de mídia em geral (debates, entrevistas, redes sociais). Para combater a desinformação, [Hassan et al. 2017] e [Nadeem et al. 2019] usam esse tipo de sistema para construir suas abordagens. ClaimBuster [Hassan et al. 2017] destaca-se como um serviço que funciona de forma totalmente automatizada, monitorando e identificando as declarações que podem ser validadas, e informando o veredicto através de um portal na Web[1] e no Twitter. Por outro lado, FAKTA [Nadeem et al. 2019] não possui um monitoramento ativo de afirmações, mas também é capaz de realizar o processo automaticamente. Para isso, os autores utilizam-se métodos de aprendizagem de máquina para identificar o posicionamento relevante do documento em relação ao que se pretende verificar.

Mesmo que esses trabalhos consigam lidar com o problema de *fact-checking*,

[1]https://idir.uta.edu/claimbuster/

eles não são capazes de prover uma arquitetura aberta e capaz de interoperar com outros métodos já existentes, ou reutilizá-los. A utilização de componentes isolados e interoperáveis ajuda a evitar desperdícios com retrabalhos e acelerar o tempo de desenvolvimento [Both et al. 2016]. Na área de Processamento de Linguagem Natural, *Natural Language Processing - NLP*, foram encontrados dois usos pra esse tipo de técnica. Em [Volodina et al. 2012], essa técnica foi aplicada para o aprendizado de idiomas, onde foi criada uma arquitetura com base em *web services* onde os serviços trocam anotações entre si. Similarmente, [Both et al. 2016] também realiza uma tarefa semelhante para prover interoperabilidade em sistemas de *Question Answering*. Esse trabalho também se apoia em uma ontologia, a QA Ontology, que é utilizada para armazenar todas as informações geradas no processo de construção da resposta.

Ainda que esses trabalhos apresentem avanços em suas respectivas áreas, eles podem não se adaptar completamente à área de *fact-checking*. Durante o processo de verificação, é preciso coletar e e processar evidências para se estabelecer um veredito. Ao organizar semanticamente essas informações, através de uma ontologia que aborda todo o processamento, faz com que seja possível capturar melhor o conhecimento disponível [Munir and Sheraz Anjum 2018]. Na literatura, os trabalhos disponíveis descrevendo ontologia ou formas de anotação de dados para *fact-checking* não contemplam esse aspecto. Trabalhos como [Rehm et al. 2018] focam em criar uma forma de descrever os resultados obtidos pela verificação, e não detalhar o processo de verificação em si. O presente trabalho, por sua vez, apresenta uma proposta de framework para abordagens de *fact-checking* que permite auxiliar na reprodutibilidade de experimentos e na construção de novas abordagens através do reuso de componentes entre abordagens distintas. Uma vez que o processo de *fact-checking* é dividido em etapas que podem ser definidas e alteradas facilmente no framework, este permite também auxiliar no processo de experimentação com diferentes componentes. Ainda, para garantir a rastreabilidade de informações do processo de verificação, é apresentada uma ontologia que descreve as relações e entidades que fazem parte do processo de validação, e que pode usada por qualquer abordagem ou sistema fora deste framework.

3. Ontologia Revera

Para fornecer interoperabilidade entre os componentes da mesma etapa, foi desenvolvida a ontologia ReVera[2] para descrever todos os dados que podem ser gerados em cada uma das etapas do processo de verificação automática de fatos à partir de dados textuais. Assim, independentemente da implementação, é possível obter uma mesma organização semântica dos dados gerados. Através desse tipo de especificação, é possível intercambiar informações entre diferentes sistemas, através da utilização de um vocabulário em comum [Bittner et al. 2006].

A PROV Ontology (PROV-O) [Belhajjame et al. 2012] foi definida para ser simples e extensível pelas aplicações que irão utilizá-la, onde a partir dessa simplificação, já é possível criar representações específicas para um determinado domínio. Sendo assim, a partir dessas classes e relações, foi definida uma ontologia de proveniência, com base na PROV-O, para definir os dados de proveniência que podem ser gerados e consumidos durante o processo de verificação das etapas, definido no *pipeline* proposto.

[2]Origina-se do latim, onde *revera* significa “na verdade”

A ontologia desenvolvida, chamada de ReVera Fact Checking Ontology, referida através do prefixo *fc:*, mapeia todas as entidades que são geradas durante o processamento. Ela foi pensada para ser uma ontologia de proveniência de dados, onde os dados gerados em um determinado estágio da *pipeline* de *fact-checking* são propriamente identificados. Isso faz com que exista um certo nível de reprodutibilidade dos resultados obtidos, podendo ser identificados e reprocessados com quaisquer outras configurações de componentes. O diagrama da ontologia pode ser observado na Figura 1. Na representação, os círculos brancos representam as classes definidas na ontologia e os azuis indicam as classes herdadas da PROV-O. As arestas indicam as relações, onde a origem é o sujeito e o alvo é o objeto, sendo as arestas azuis relações herdadas, as vermelhas, definidas pela ReVera Ontology, e as pretas não nomeadas na figura indicam relações do tipo *is a* (é um/é uma).

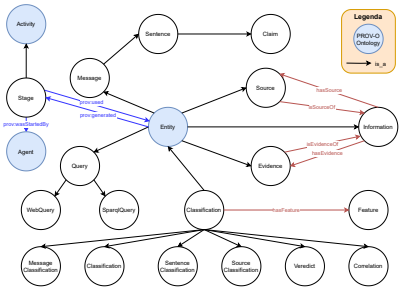


Figure 1. Representação de classes e relações da ontologia proposta

A ontologia proposta estende a PROV-O, especializando classes como Entidades e Atividades especificamente para representar o *pipeline* de processamento das afirmações.É importante destacar que, em uma abordagem de pesquisa recente, a FC Ontology foi desenvolvida no editor de ontologias Onto4ALLEditor[3], que possibilita a construção colaborativa de ontologias por mais de um usuário através da web e faz uma validação dos componentes ontológicos (classes, relações, propriedades e axiomas) construídos [Mendonça et al. 2020]. Essa característica atribuída à construção da ReVera Ontology contribuiu com a qualidade de seu conteúdo ontológico.

4. ReVera Framework

A fim de fornecer uma plataforma para o desenvolvimento de abordagens para a verificação de fatos, o *framework* foi desenvolvido com base nas etapas necessárias para serem implementadas em sistemas *end-to-end*, o ReVera Framework. Além disso, ele foi desenvolvido de forma a permitir o reuso, ser extensível e paralelizável.

Essa arquitetura necessita de um módulo de gerenciamento central para receber e gerenciar todas as solicitações de processamento. Cada uma das implementações das etapas possíveis, chamados de componentes, precisam se comunicar com o *core*, já que é necessária a entrada e saída de dados entre eles. Essa comunicação, entre os componentes e o core são realizadas através de um mensageiro. A Figura 2 mostra a arquitetura proposta.

[3]http://onto4alleditor.com/

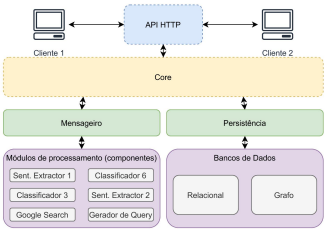


Figure 2. Arquitetura proposta para o *ReVera Framework*

O *core* é o módulo que precisa ter visão de todo o sistema. Ele irá receber todas as solicitações de processamentos, além de fornecer uma interface para a comunicação externa para se obter os resultados gerados pelo processamento. Para manter todas as informações, ele se comunica diretamente com dois bancos de dados específicos: um relacional e outro orientado a grafos. A base de dados relacional mantém o registro de todas as requisições de processamento, além de rastrear todos os componentes necessários para o processamento da requisição, identificando quais já foram concluídos e quais ainda precisam serem executados. Os dados gerados pela execução de cada etapa, descritos pela ontologia, são organizados em um grafo e salvos em um banco de dados na forma de triplas RDF (*Resource Description Framework*). Esse módulo é necessário para o sistema evitar que os componentes que realizam o processamento tenham que lidar com o encaminhamento e o controle de estado de cada uma das requisições.

Os componentes precisam ser especializados para realizar a tarefa necessária para que a etapa seja concluída. Ao ligar, em cadeia, componentes de diferentes etapas é possível transformar as sentenças ou triplas de entrada, encontrando os dados que o classificador final necessita para avaliar sua veracidade. Essas peças conectam-se ao núcleo através de mensageiros, onde para cada um dos componentes é criado uma fila de mensagens. Caso haja mais de uma instância de um mesmo componente, esses componentes escutam a mesma fila e isso permite paralelizar as mensagens que são inseridas na fila. A utilização desse tipo de arquitetura, muito utilizada em conjunto com microsserviços, permite um desacoplamento grande entre todos os módulos do *framework*, pois os módulos não precisam tratar da comunicação diretamente, não havendo restrições para a aplicação de diferentes plataformas ou linguagens de programação.

Ao ser implementada dessa forma, o *framework* apoia-se na ontologia ReVera para mediar o funcionamento de todos os componentes. Essa característica é o que cria uma padronização semântica entre todas as entradas e os resultados dos processamentos realizados. Isso acaba facilitando características como reuso, através da reutilização dos componentes, além de reprodutibilidade e rastreabilidade, que permitem que os resultados possam ser compreendidos e reproduzidos através das informações geradas durante o processo.

Nesse contexto, existe um desacoplamento entre os componentes, já que toda execução fica restrita a coerência entre os tipos de entrada e saída esperados para cada um dos componentes. Isso permite que a substituição de um componente por outro do mesmo tipo seja menos custosa, e que a sua reutilização para criação de novos *pipelines* seja mais simples, já que basta apenas considerar suas entradas e saídas.

Todos esses artefatos gerados durante o processamento são persistidos no banco de dados, permitindo que todo o fluxo de troca de informações possa ser analisado posteriormente. Uma das implicações diretas é a possibilidade de tornar o processo de reprodutibilidade mais transparente, podendo explicitar os dados que são consumidos e gerados. Desse modo, é possível consumir os dados de entrada para verificar se é possível obter tais dados, comparando diretamente com os resultados que foram obtidos.

A segunda implicação é permitir que os dados gerados possam ser rastreados, podendo analisar a transformação dos dados entre as etapas. Consequentemente, é possível obter todas as evidências e artefatos utilizadas para validar uma informação, desde a requisição até a finalização do processamento. Portanto, métodos que trabalham com a geração da explicação de forma automática podem se beneficiar disso como uma forma de obtenção dos dados.

Além disso, ao utilizar a estrutura já fornecida, são disponibilizados métodos para o gerenciamento e armazenamento dos fluxos de execução, utilizando um vocabulário pré-definido, podendo simplificar o uso dos dados de saída para diferentes sistemas.

5. Validação

Visando servir como uma prova de conceito para o *framework*, foi desenvolvido um *pipeline* baseado no mesmo trabalho anterior [Gerber et al. 2015]. Todas as etapas de processamento foram dividas em componentes de responsabilidade única dentro do *framework*, onde cada componente está responsável por uma forma de implementação da etapa de processamento. Além dos componentes que puderam ser identificados a partir de [Gerber et al. 2015], foram adicionados outros a fim de alterar a forma de processamento para ter como entrada documentos em texto de linguagem natural, ao invés de triplas RDF. Para isso, foi necessário alterar as etapas iniciais de tratamento da entrada.

Assim que a resposta é produzida, é responsabilidade de cada componente informar ao *core* o documento de resposta. Essa resposta é feita através do mensageiro, por uma fila específica, contendo o identificador do grafo, necessário para agrupar todos os dados relacionados a requisição e o próprio documento resultante. Assim, ao receber esses dados, o *core* é capaz de persistir os dados e prosseguir com o processo, identificando a próxima etapa e enviando todos os dados persistidos no grafo da requisição como entrada.

Analogamente, as outras etapas possuem protocolos de entrada e saída similares: todas procuram por instâncias de certas classes no documento de entrada e, ao finalizarem o processamento, gerar uma resposta seguindo o vocabulário da ReVera Ontology. Com isso, houve uma reutilização dos componentes já implementados e, através de uma reorganização do fluxo de execução, foi possível experimentar uma abordagem diferente.

Dessa forma, utilizando o *framework*, os componentes não precisaram lidar com os possíveis fluxos de dados que possam ocorrer, limitando-se apenas a compreender o vocabulário de suas entradas e saídas. Isso facilitou a extensão do trabalho de [Gerber et al. 2015] de maneira com que foi possível reaproveitar grande parte dos códigos já implementados.

6. Conclusões e Trabalhos Futuros

Este trabalho apresenta um *framework* e uma ontologia para rastreabilidade, a ReVera Ontology, para o desenvolvimento de métodos de verificação automática de fatos (*fact-checking*). O *framework* consiste de um *core* e módulos que implementam determinadas partes do processamento (componentes), é capaz de receber solicitações de processamento de diferentes *pipelines* e montar um grafo com informações de proveniência, seguindo o vocabulário da ontologia.

Além disso, o desenvolvimento do *framework* proposto nesta pesquisa pode ser considerado uma solução inédita para a área de *fact-checking*. Foram identificados na literatura trabalhos para *Question Answering* e *Language Learning*, porém não foram encontradas soluções desse tipo para *fact-checking*. A implementação forneceu uma forma desacoplada e distribuída para implementar sistemas de verificação, permitindo uma maior facilidade para alterar individualmente as partes do processo. Além disso, através do *framework*, foi possível entender o funcionamento de diversas pesquisas da área, gerando também uma classificação para as principais etapas realizadas durante o processamento.

Como trabalhos futuros, pretende-se implementar mais componentes para o *framework*, buscando novos trabalhos disponíveis na literatura e aplicando-os para que possam utilizar a ontologia. Novos componentes permitem a criação de novos *pipelines*, o que permite a evolução, verificando deficiências para mais casos de testes.

Além disso, devido a possibilidade de serem testadas diferentes combinações de componentes, futuramente podem ser adicionadas ferramentas de avaliação para os *pipelines* desenvolvidos sobre o *framework*. Elas devem permitir executar diferentes componentes e verificar, através de métricas de performance disponíveis na literatura, o desempenho de diferentes fluxos em um conjunto de testes, auxiliando em metodologias de experimentação dos resultados.

References

Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., and Zhao, J. (2012). Prov-o: The prov ontology. Technical report.

Bittner, T., Donnelly, M., and Winter, S. (2006). Ontology and semantic interoperability.

Both, A., Diefenbach, D., Singh, K., Shekarpour, S., Cherix, D., and Lange, C. (2016). Qanary — a methodology for vocabulary-driven open question answering systems. In *Proceedings of the 13th International Conference on The Semantic Web. Latest Advances and New Domains - Volume 9678*, page 625–641, Berlin, Heidelberg. Springer-Verlag.

Conroy, N. J., Rubin, V. L., and Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

da Silva, F. R. M., Freire, P. M. S., de Souza, M. P., de A. B. Plenamente, G., and Goldschmidt, R. R. (2020). Fakenewssetgen: A process to build datasets that support comparison among fake news detection methods. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, WebMedia '20, page 241–248, New York, NY, USA. Association for Computing Machinery.

Gerber, D., Esteves, D., Lehmann, J., Bühmann, L., Usbeck, R., Ngonga Ngomo, A.-C., and Speck, R. (2015). Defacto - temporal and multilingual deep fact validation. *Web Semantics: Science, Services and Agents on the World Wide Web*.

Graves, D. (2018). Understanding the promise and limits of automated fact-checking.

Hassan, N., Adair, B., Hamilton, J., Li, C., Tremayne, M., Yang, J., and Yu, C. (2015). The quest to automate fact-checking. *Proceedings of the 2015 Computation + Journalism Symposium*.

Hassan, N., Zhang, G., Arslan, F., Caraballo, J., Jimenez, D., Gawsane, S., Hasan, S., Joseph, M., Kulkarni, A., Nayak, A. K., Sable, V., Li, C., and Tremayne, M. (2017). Claimbuster: The first-ever end-to-end fact-checking system. *Proc. VLDB Endow.*, 10(12):1945–1948.

Kotonya, N. and Toni, F. (2020). Explainable automated fact-checking for public health claims.

Mendonça, F. M., de Castro, L. P., de Souza, J. F., Almeida, M. B., and Felipe, E. R. (2020). Onto4alleditor: um editor web gráfico de ontologias direcionado a diferentes tipos de desenvolvedores de ontologias. *Proceedings of the XIII Seminar on Ontology Research in Brazil and IV Doctoral and Masters Consortium on Ontologies (ONTOBRAS 2020)*.

Miranda, S., Nogueira, D., Mendes, A., Vlachos, A., Secker, A., Garrett, R., Mitchell, J., and Marinho, Z. (2019). Automated fact checking in the news room. *CoRR*, abs/1904.02037.

Munir, K. and Sheraz Anjum, M. (2018). The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics*, 14(2):116–126.

Nadeem, M., Fang, W., Xu, B., Mohtarami, M., and Glass, J. (2019). Fakta: An automatic end-to-end fact checking system.

Rehm, G., Schneider, J. M., and Bourgonje, P. (2018). Automatic and manual web annotations in an infrastructure to handle fake news and other online media phenomena. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Santos, R. and Pardo, T. (2020). *Fact-Checking for Portuguese: Knowledge Graph and Google Search-Based Methods*, pages 195–205.

Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.

Souza, J., Gomes Jr, J., Marques, F., Julio, A., and Souza, J. (2020). A systematic mapping on automatic classification of fake news in social media. *Social Network Analysis and Mining*, 10.

Volodina, E., Borin, L., Loftsson, H., Arnbjörnsdóttir, B., and Leifsson, G. Ö. (2012). Waste not, want not: Towards a system architecture for icall based on nlp component re-use. volume 80, pages 47–58.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

An Empirical Study of Information Retrieval and Machine Reading Comprehension Algorithms for an Online Education Platform

Eduardo F. Montesuma[1], Lucas C. Carneiro[1], Adson R. P. Damasceno[2], João Victor F. T. de Sampaio[1], Romulo F. Férrer Filho[1], Paulo Henrique M. Maia[2], Francisco C. M. B. Oliveira[2]

[1]Federal University of Ceará
Fortaleza – CE – Brazil

[2]State University of Ceará
Fortaleza – CE – Brazil

{firstname.lastname@dellead.com}

Abstract. *This paper provides an empirical study of various techniques for information retrieval and machine reading comprehension in the context of an online education platform. More specifically, our application deals with answering conceptual students questions on technology courses. To that end we explore a pipeline consisting of a document retriever and a document reader. We find that using TF-IDF document representations for retrieving documents and RoBERTa deep learning model for reading documents and answering questions yields the best performance with respect to F-Score. In overall, without a fine-tuning step, deep learning models have a significant performance gap with comparison to previously reported F-scores on other datasets.*

1. Introduction

In distance learning courses, tutors play a crucial role due to performing several activities, such as pedagogical support, student performance, interaction monitoring, dropout detection, prevention, and reduction [Barker 2002, Denis et al. 2004], thus helping students to finish their course successfully [Simpson and Sharma 2002, Lentell 2004]. However, when a large number of students attend courses, human tutors can be overloaded, which may have a negative impact on their work. Bernath and Rubin (2001) report how the sheer volume of online activities can be too much for the teacher and the student and why the workload on online teachers is often reported to be significantly greater than what it is in a face-to-face teaching context.

In this realm, Damasceno *et al.* (2020) previously proposed a chat-bot called STUART for easing the burden in distance learning courses. It uses Natural Language Processing (NLP), machine learning techniques, and interaction with Dell Accessible Learning (DAL) [1] learning tools for responding to student's pedagogical, technical, and content demands. [Damasceno et al. 2020] does so by sending proactive pedagogical recommendations according to the student's profile and for ensuring the reduction of activities that require pedagogical resources and human tutoring. In this work we focus on content responses using the Question Answering (QA) technology.

[1]http://leadfortaleza.com.br/dal/

As follows, QA technology is the possible solution to mitigate that problem [Wen et al. 2012]. A QA system aims to automatically answer some of the students questions, thus reducing the workload on teachers. Modern QA systems are composed by various sub tasks. In this work we focus on two of then: document retrieval and document reading, under the perspective of NLP. Concerning the document retrieval task, it consists of finding the document that is most similar to a given question. This is done either by calculating the similarity between documents, through the usage of document representations as Term Frequency-Inverse Document Frequency (TF-IDF) [Jones 1972] or by word representations [Pennington et al. 2014].

After finding the appropriate document, one needs to extract the answer. A possible approach task that gained attention in the literature is called Machine Reading Comprehension (MRC) [Hermann et al. 2015], which aims at teaching machines to answer questions after comprehending given passages or contexts. The state-of-the art in MRC are deep language models based on attention mechanism, such as BERT [Devlin et al. 2019], ELECTRA [Clark et al. 2020] and ALBERT [Lan et al. 2020]. Despite the many advances in the field, the usage of these tools remains data intensive. For instance, when applying BERT for a domain-specific QA, such as biology, a fine-tuning step is necessary since the word distribution can drastically change across domains [Lee et al. 2020]. This is an important drawback, specially for small to medium-sized distance learning platforms that do not have a reasonable corpus for fine-tuning the model.

Due to the wide variety of strategies for performing retrieval and reading tasks, choosing the appropriate model for each of them is cumbersome. In this sense, the majority of work evaluate either the retrieval task, or the reading performances alone. Nevertheless, some surveys provide a broader view on the subject. For instance, in [Abbasiantaeb and Momtazi 2021], the authors provide a comprehensive review of the complete QA pipeline, covering various deep learning approaches. Moreover, Fu et al. (2020) covers traditional methods, such those that rely on predefined rules or templates for answering questions.

In this context, our work differentiates itself from these approaches since it provides an empirical study of deep learning-based QA in the context of online distance courses, predominantly with technology subjects. Therefore we present an empirical study of retrieval and reading tasks working in consonance. Our study is centered around an existing educational platform, the DAL platform. We investigate the following questions: (i) Which algorithms better fit the pipeline used for QA? (ii) Can pre-trained deep learning models perform as well even when fine-tuning is not possible? (iii) Can deep learning models be used for answering conceptual questions of the DAL platform students?

The remainder of this paper is organized as follows. Section 2 gives an overview of the proposed methodology with Document Retriever and Document Reader. Section 3 presents the results and inferences obtained from them. Section 4 draws the conclusions we can draw from our work.

2. Methodology

In this section we explore the methodology for QA. In particular, we propose using a pipeline similar to that of [Abbasiantaeb and Momtazi 2021], with slight modifications. It is composed by three elements: (i) a corpus of contexts, (ii) a document retriever module, and (iii) a document reader module. An overview on how these components work in consonance is shown in Figure 1.

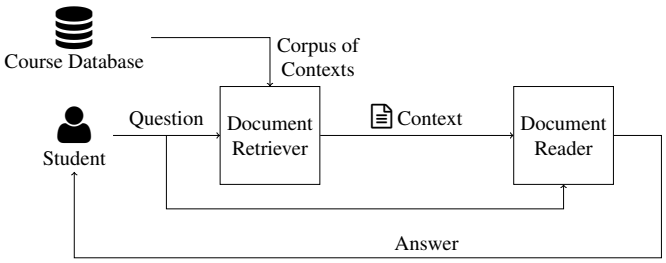


Figure 1. Pipeline for the MRC module.

2.1. Corpus of Contexts

We create a usable knowledge base for answering the students questions by gathering textual content from courses offered in the DAL platform. The DAL platform provides a variety of multidisciplinary online courses, mostly about technology or management. The lectures of the courses are displayed as web pages or videos.

The data collection process was made through an automatic script that downloads all the content from DAL's platform courses. For online lectures, we parsed the HTML files to extract the textual content of web lectures. For video lectures, we extracted the textual content from subtitles. In total, we gathered 674 documents, from 189 lectures of 18 courses as illustrated by Figure 2.

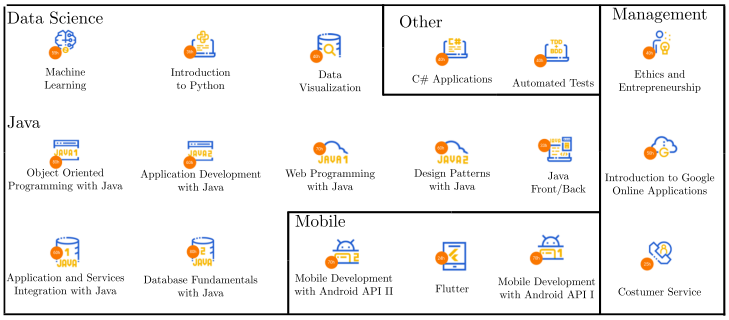


Figure 2. An overview of the 18 courses available in the DAL's platform.

After obtaining the texts, we split the documents into segments consisting of a maximum of $T = 128, 256,$ and 512 tokens, resulting in three knowledge bases with the same content, but divided in distinct number of segments. In addition, we executed prepossessing steps in the texts, which include normalization in lower case, removing stop words, punctuation, numbers and accentuation, and lemmatisation.

Furthermore, we also created a corpus of conceptual questions for evaluating our pipeline. For the 10 most popular courses, DAL's tutors created questions based on their past experience of what students may ask. For each test question, the following labels are available: the course and lecture related to the question, the context that answers the question, and the desired answer within the context. This allows us to evaluate both the retrieving and reading tasks.

2.2. Document Retriever

The Document Retriever component executes an Information Retrieval task [Kolomiyets and Moens 2011] by retrieving from a corpus a document that is expected to contain the answer for a question, i.e. the context of the question. The criteria to find this context is to compute a similarity measure between the question and the documents in a given vector representation, in which the document with a higher similarity is chosen. The underlying hypothesis is that the greater the similarity between the question and a document, the more likely that document contains the answer to the question. In a distance learning application, each document refers to an excerpt of the textual content of a lecture from an online course. Besides the text of the candidate document, the name of the course and the lecture identifier is also returned.

There are various approaches for evaluating the similarity between documents and questions. In this work we focus on those techniques involving the numerical representation of documents. As follows, one tries to represent either the entire document, or its words by a vector $\mathbf{x} \in \mathbb{R}^n$. Examples of the former approach are the so-called Bag of Words (BoW) and TF-IDF [Jones 1972], which roughly rely on the word frequency in each document. Moreover, Global Vectors (GloVe) word embeddings [Pennington et al. 2014], which are trained on large corpus of texts in an unsupervised fashion is an example of the latter approach.

Once one has a representation for the document, it is still necessary to choose a notion of distance between the vectors. A common choice in the literature is using the cosine similarity. On the other hand, when using word embeddings, novel distances such as the Word Mover Distance (WMD) [Kusner et al. 2015] can be used. The WMD is particularly interesting since it considers documents as empirical probability measures over the word space.

2.3. Document Reader

The Document Reader component works as a MRC module. It implements a Deep Learning model to predict possible answers to questions made by students using the DAL platform. The Document Retriever module provides a context from which the Document Reader extracts an answer to those questions. The prediction itself is made by finding the beginning and the end of the answer in the context provided by the Document Retriever. It uses ELECTRA as its deep learning model, but BERT, RoBERTa, and ALBERT were also used for comparison purposes, and are briefly described below.

Bidirectional Encoder Representation from Transformers (BERT) [Devlin et al. 2019] is a Masked Language Model (MLM) [Taylor 1953] that selects a small subset of the unlabeled input data and masks its tokens identity (15% of tokens are masked, replaced, or left unchanged at random.) The network is then

trained to predict the original input. It achieved state-of-the-art results in various NLP tasks, such as QA and Sentiment Analysis. One of BERT's major innovations is applying bidirectional training to language representations as opposed to single direction (usually left-to-right modeling or a combination of both left-to-right and right-to-left modeling) training. It allows the fusion of both contexts, the one to the left of the masked token and the one to the right, to predict the masked word found in the original input. BERT also uses a next sentence prediction task that can capture the relationship between sentences.

The Robustly optimized BERT approach (RoBERTa) [Liu et al. 2019] improves on BERT by optimizing its method of pretraining. The authors evaluated how hyperparameter tuning and training set size impact the performance of BERT-like models. The model is trained longer, with larger batches and learning rates, on more data. The model excludes the next sequence prediction task and it is trained on longer sequences. Also, the model has a dynamic masking strategy applied to the training data, where a masking pattern is generated every time a sequence is fed to the model. The model itself is a reimplementation of BERT with the aforementioned modifications and improved on BERT's results on General Language Understanding Evaluation (GLUE) [Wang et al. 2018] and Stanford Question Answering Dataset (SQuAD) [Rondeau and Hazen 2018] benchmarks.

The model A Lite BERT (ALBERT) [Lan et al. 2020] is a low memory consumption model similar to BERT. It has two parameter reduction techniques that lower its memory consumption and increase its training speed. The first technique consists in decomposing the embedding matrix into two smaller ones, so it is easier to grow the model hidden size without increasing the number of parameters. The second one is sharing parameters across all layers to prevent the number of parameters from growing with network depth. Both techniques improve parameter efficiency without reducing model performance.

Efficiently Learning an Encoder that Classifies Token Replacement Accurately (ELECTRA) was first proposed in [Clark et al. 2020]. The model detects replaced tokens instead of recovering the original input. The input data is corrupted by replacing tokens with some others generated by a small masked language model. The network is then pretrained as a discriminator that predicts if every token is in the original input or not and can be fine-tuned on downstream tasks. ELECTRA's greatest advantage over MLM is that it learns from all input tokens instead of only from a small subset of the original data, which makes it computationally more efficient.

3. Results

In this section we describe three experiments, and discuss our results. Section 3.1 explores the space generated by the TF-IDF representation. Section 3.2 presents a comparative study of information retrieval and deep learning-based MRC algorithms. Finally, Section 3.3 provides a broad discussion on the results we obtained.

3.1. Corpus Visualization

Based on the pre-processed text from the corpus, we built the TF-IDF representation consisting on a matrix $\mathbb{R}^{N \times M}$, for N, the number of segments, and $M = 18,233$, the vocabulary size. To further explore this representation of the corpus, we perform two dimensionality reduction steps: (i) we apply Principal Component Analysis (PCA) to the

data, reducing it to a reasonable dimension (100), (ii) from the reduced vector, we apply t-distributed Stochastic Neighbor Embeddings (t-SNE) [Van der Maaten and Hinton 2008] to visualize the data on $\mathbb{R}^2$, as shown in Figure 3.

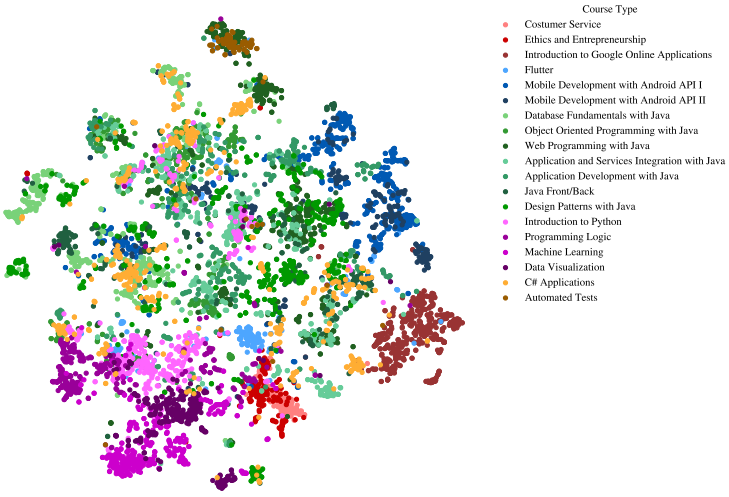


Figure 3. t-SNE embedding of the TF-IDF representation of documents. The colours represent each the 18 courses in the DAL platform.

3.2. Pipeline Evaluation

The evaluation of the pipeline shown in Figure 1 is carried out in two steps. First, the information retrieval task is evaluated, according to both course and lecture accuracy. It is important to note that a lecture prediction is only correct if the document retriever also predicts the course correctly. At this step, we also compare different choices for document segmentation, as we evaluate each candidate retrieval algorithm for a maximum of 128, 256 and 512 tokens.

Thus, we compare 4 different approaches for retrieving documents: (i) the TF-IDF representation [Jones 1972], (ii) the usage of BoW, both using cosine similarity (iii) Portuguese GloVE Word Embeddings [Hartmann et al. 2017] using the so-called WMD and (iv) using Word Centroid Distance (WCD), both as proposed by [Kusner et al. 2015], which reported that WCD outperformed both BoW and TF-IDF in tasks of Information Retrieval by significant margins in various datasets. The comparison is shown in Table 1.

Furthermore, for the comparison of MRC methods presented in Section 2.3, we use the F-Score, a metric previously used in [Rajpurkar et al. 2016] for QA. Based on this metric, we present two comparisons. First, we consider that the answer context is known *a priori*, thus yielding a similar evaluation to previous studies, but in a different context (DAL database). Second, we consider the evaluation of the pipeline as a whole. In this case, the context is predicted using the best information retrieval method, which is TF-IDF with a segmentation of 128 tokens, as presented in Table 1. The MRC results are shown in Table 2.

	Course Accuracy			Lecture Accuracy		
Feature Choice	128	256	512	128	256	512
GloVE Embeddings						
- WMD	38.18	33.66	37.12	26.73	23.76	22.77
- WCD	35.15	34.16	31.18	23.76	17.82	13.86
BoW	50.49	42.08	42.08	39.11	32.17	32.17
TF-IDF	72.77	65.84	63.86	62.37	55.94	52.97

Table 1. Comparison of Information Retrieval methods based on course and lecture accuracy.

Model	# Parameters	Context Known	Context Retrieved
BERT	334M	13.39	9.22
ALBERT	235M	15.42	14.16
ELECTRA	334M	15.81	16.62
RoBERTa	354M	15.88	16.83

Table 2. Comparison of MRC algorithms based on F-Score, in two scenarios: (i) context is known *a priori*, and (ii) context is retrieved by the document retriever.

3.3. Discussion

Figure 3 evidences the difficulty of detecting the appropriate course for a given question, as the course's content can be highly multi-disciplinary. For instance, the extreme top cluster of Figure 3 is composed by documents from "Web Programming with Java" and "Automated Tests". At a first glance, these two courses are unrelated, but the documents treat the same subject (in the former course, it treats tests in Java).

Another source of confusion is related to the courses "Machine Learning", "Data Visualization" and "Introduction to Python", at the bottom of Figure 3, in shades of purple. Even though the first 2 of these 3 courses treat advanced topics, they are taught using Python, thus there is an intersection in their content, mostly in the initial lectures. This is also the case for the course "C# Applications", in yellow. The course content involves, among other topics: object-oriented programming, databases, and web applications. Consequently, the documents of this course are scattered throughout the t-SNE embedding. Hence, the Information Retrieval task achieved better results for all methods with the segmentation of 128 tokens, indicating that short texts are more easily distinguished. This may happen since these have less possibility of overlapping topics.

As a possible solution for this issue, we may leverage the DAL platform by retrieving which courses a given student is enrolled in. This allows us to narrow the search to a few courses. Even though a student may be enrolled in courses that share content, the system will likely perform better in terms of document and lecture accuracy.

Secondly, unsupervised document representation approaches, such as BoW and TF-IDF have superior performance than GloVe embeddings. This was expected, since the domain of our application is very specific. More specifically, Hartmann *et al.*

(2017) trained the embeddings on a corpus consisting mainly on Wikipedia and news pages, whereas our application is mainly composed of technology-related courses. As a consequence, training and test data follow different probability distributions, harming the test performance. A possible solution that may improve the performance of GloVe is training on a corpus that is more similar to our application, or performing fine-tuning. However, due to the limited size of the DAL's corpus, this was not feasible.

Thirdly, in comparison with the results reported on the respective paper of each model, there is still a wide margin improvement. For instance, Clark *et al.* (2020) report a F-Score of 94.9 on SQuAD v1.1 dataset, whereas in the DAL test set the performance is much lower (16.83 maximum). The reason for this performance gap is, again, the distributional shift between training and test sets for these models. Possible approaches for solving this issue are: re-training, fine-tuning or even performing transfer learning [Ganin et al. 2016].

Finally, note that as reported in Table 2, RoBERTa and ELECTRA have slightly better performance when the context is retrieved through the document retriever. This is mainly due the fact that different contexts may contain the answer for a given question. This also highlights that the context indicated by the tutors may not be the most appropriate for the model to extract the answer from.

4. Conclusion

This work presented an empirical study of information retrieval and machine reading comprehension algorithms for an online education platform. We used a pipeline consisting of a corpus of contexts based on courses content, a document retriever, and a document reader. The pipeline has two challenges to overcome. First, contexts are not always informative. They might be guessed wrongly by the document retriever, or the information may be scattered across documents. Secondly, the Language Model (LM), which have been built using general language and broad scope texts, are applied to a corpus that is very content specific (technology courses, for instance). This hinders the quality of answers.

Concerning the best choices for the QA pipeline, using unsupervised document representations, such as TF-IDF, yields better performance than pre-trained word embeddings, such as GloVe. Among these, the former has the better retrieval performance. Furthermore, when comparing MRC algorithms, ELECTRA and RoBERTa give the best results, having similar performance. Moreover, the overall pipeline, that is, TF-IDF retriever and RoBERTa-based reader with a document segmentation of 128 tokens, yields slightly better answers than using the reader with contexts provided by the DAL tutors. Furthermore, without fine-tuning, there is a considerable gap in F-Score when using pre-trained deep learning-based MRC algorithms in a specific domain QA problem, such as answering conceptual questions about technology courses.

Given the shortcomings of applying pre-trained deep learning-based models in specific contexts, deep learning models can be used for answering conceptual questions. Future work involve: (i) improving the DAL's data base for allowing the fine-tuning of MRC algorithms, and (ii) employing transfer learning for efficiently using the data at our disposal for improving pre-trained models.

References

Abbasiantaeb, Z. and Momtazi, S. (2021). Text-based question answering from information retrieval and deep neural network perspectives: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1412.

Barker, P. (2002). On being an online tutor. *Innovations in Education and Teaching International*, 39(1):3–13.

Bernath, U. and Rubin, E. (2001). Professional development in distance education – a successful experiment and future directions. *Innovations in Open & Distance Learning, Successful Development of Online and Web-Based Learning*, pages 213–223.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Damasceno, A. R., Martins, A. R., Chagas, M. L., Barros, E. M., Maia, P. H. M., and Oliveira, F. C. (2020). Stuart: an intelligent tutoring system for increasing scalability of distance education courses. In *Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems*, pages 1–10.

Denis, B., Watland, P., Pirotte, S., and Verday, N. (2004). Roles and competencies of the e-tutor. In *Networked Learning 2004: A Research Based Conference on Networked learning and lifelong learning: Proceedings of the fourth international conference, Lancaster*, pages 150–157.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fu, B., Qiu, Y., Tang, C., Li, Y., Yu, H., and Sun, J. (2020). A survey on complex question answering over knowledge base: Recent advances and challenges. *arXiv preprint arXiv:2007.13069*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Hartmann, N. S., Fonseca, E. R., Shulby, C. D., Treviso, M. V., Rodrigues, J. S., and Aluísio, S. M. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In *Anais do XI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 122–131. SBC.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems*, 28:1693–1701.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Kolomiyets, O. and Moens, M.-F. (2011). A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966. PMLR.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Lentell, H. (2004). The importance of the tutor in open and distance learning. In *Rethinking Learner Support in Distance Education*, pages 76–88. Routledge.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Rondeau, M.-A. and Hazen, T. J. (2018). Systematic error analysis of the stanford question answering dataset. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 12–20.

Simpson, O. and Sharma, R. C. (2002). Book review-supporting students in open and distance learning. *International Review of Research in Open and Distance Learning*, 3(3).

Taylor, W. L. (1953). "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11):2579–2605.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Wen, D., Cuzzola, J., Brown, L., and Kinshuk, D. (2012). Instructor-aided asynchronous question answering system for online education and distance learning. *International Review of Research in Open and Distributed Learning*, 13(5):102–125.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Assessing the Impact of Stemming Algorithms Applied to Brazilian Legislative Documents Retrieval

Ellen Souza[1,2], Gyovana Moriyama[2], Douglas Vitório[1,3], André C. P. L. F. de Carvalho[2], Nádia Félix[2,4], Hidelberg O. Albuquerque[1,3], Adriano L. I. Oliveira[3]

[1]MiningBR Research Group, Federal Rural University of Pernambuco
CEP: 52171-900 – Recife/PE – Brazil

[2]Institute of Mathematics and Computer Sciences, University of São Paulo, Brazil

[3]Centro de Informática, Federal University of Pernambuco, Brazil

[4]Institute of Informatics, Federal University of Goiás, Brazil

ellen.ramos@ufrpe.br, gymori@usp.br, andre@icmc.usp.br

{damsv,hoa,alio}@cin.ufpe.br, nadia.felix@ufg.br

Abstract. *The main purpose of stemming is to reduce the inflected words into its root form or stem. Thus, words can be mapped to the same concept, improving the process of information retrieval, regarding its ability to index documents and to reduce data dimensionality. However, the efficiency of those algorithms varies according to different aspects. Also, studies in the field area reached contrasting conclusions. This work assesses the use of stemmers in the retrieval of legislative documents written in Portuguese. Four stemmers together with BM25 were evaluated in two legislative corpora from the Brazilian Chamber of Deputies. RSLP-S and Savoy stemmers showed the best improvements in the information retrieval pipeline.*

1. Introduction

Information retrieval (IR) looks for unstructured material from within large collections, satisfying an information need [Manning et al. 2008]. Law was one of the first knowledge areas to adopt IR, with the first domain-specific legal retrieval system appearing as early as 1960 [Maxwell and Schafer 2008]. The importance of legal applications created a sub-area of IR, *Legal IR*, which covers a large variety of legal texts, including legislation, case law, and scholarly works [Maxwell and Schafer 2008].

In the last years, due to the huge amount of information available, which continues to increase rapidly, improved IR techniques have become necessary [Moral et al. 2014]. Thus, stemming algorithms, which can generate concise word representations, has been largely used in information retrieval systems [Alvares et al. 2005]. When used for IR, stemming can improve predictive accuracy and reduce computational costs, being one of the first steps in the IR pipeline [Moral et al. 2014, de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018].

A stemming algorithm, also called stemmer, extracts the morphological root, stem, of a word. For such, a stemmer removes affixes that carry grammatical or lexical information about the word [Moral et al. 2014]. A stemmer can: (i) cluster words according

to their topic, as many words are derivations from the same stem and can be considered as belonging to the same concept; (ii) index the documents in an IR process, according to their topics, as their terms are grouped by stems (that are similar to concepts), and (iii) reduce the collection of documents to a set of topics or stems, which can both reduce the space needed to store the structures used by an IR system and the computational load [Moral et al. 2014].

The efficiency of stemmers varies according to the language used with and the application domain [Alvares et al. 2005]. Studies evaluating the effects of stemming for IR reached contrasting conclusions [Orengo and Huyck 2001]. Researchers compared well-known stemming algorithms for texts in the English language and did not find any significant improvement due to the use of stemming, with an increase in recall and reduction in precision [Orengo and Huyck 2001]. However, authors agreed on its benefits in specific contexts, such as when the language is highly inflective (the case of the Portuguese language), when documents are short or when there is limited space for storing data [Alvares et al. 2005]. Some researchers also argue that the nature of the documents can influence its predictive performance [Alvares et al. 2005].

This work investigates the effect of the use of stemmers in the retrieval of legislative documents written in the Brazilian Portuguese language. It also investigates how the predictive performance in an IR system is affected by using dimensionality reduction techniques. To the best of the authors knowledge, this is the first evaluation of Portuguese stemming algorithms using texts from the legislative domain. The reported research is part of the *Ulysses* project, an institutional set of artificial intelligence initiatives to increase transparency, improving the Brazilian Chamber of Deputies relationship with the Brazilian population, and supporting the legislative activity [Almeida 2021].

This paper is organized as follows: Sec. 2 presents the major related studies. Sec. 3 details the IR pipeline used in this study. Sec. 4 presents the experiments performed and discusses the obtained results. Sec. 5 brings the conclusion and points out future works.

2. Related Work

We found few papers investigating the application of stemming for IR, specifically for the Portuguese language. In [Orengo et al. 2006], the authors evaluated three stemming algorithms for texts in Portuguese: Porter, RSLP, and RSLP-S. According to the experimental results, RSLP-S was the best algorithm in terms of MAP (Mean Average Precision) and Pr@10 (Precision at 10 documents). Experiments were carried out with CLEF 2006's dataset, which contains texts from Público and Folha de São Paulo newspapers.

In [Flores et al. 2010, Flores and Moreira 2016], the authors evaluated the benefits of stemming in four different languages, one of them Portuguese. They compared 8 stemmers, 5 specifically designed for the Portuguese language (Porter, RSLP, RSLP-S, Savoy, and StemBR) and 3 language-independent ones (Linguistica, GRAS, and Stemmer-S). The algorithms were evaluated in two different ways: 1) by their quality, using the Paice's method, which uses the metrics of Overstemming Index (OI), Understemming Index (UI), Stemming Weight (SW), and Error Rate Relative to Truncation (ERRT); and 2) by their impact on document retrieval effectiveness. For the Portuguese language, in the first evaluation, the best algorithms, using ERRT, were RSLP and Porter. In the second evaluation, using MAP, Savoy presented the best predictive performance. All the experiments used

datasets from the CLEF's tracks of 2005 and 2006, whose Portuguese corpus contains articles from the Brazilian newspaper Folha de São Paulo.

Using jurisprudential data from the Supreme Court of the State of Sergipe, the experiments reported in [de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018] also used Porter, RSLP, RSLP-S, and Savoy for two stemming evaluations. They considered the average number of unique terms obtained by each stemmer and its average percentage of reduction, in which the RSLP proved to be the best. The stemmer's impact on the legal document retrieval was evaluated in terms of MAP, MPC (average of Pr@10), and MRP (average of R-Precision) and using the BM25 IR algorithm. The best algorithms for this task were RSLP-S and Savoy, as they reduced the dimensionality of the data and increased the effectiveness of Information Retrieval. However, the authors pointed out that the use of radicalization usually deteriorated the Okapi BM25 performance.

Although this last study focuses on the legal domain, the texts used are from judgments and monocratic decisions of Appeals Court, and judgments and monocratic decisions of Special Courts, which differ in size, entities, and vocabulary from the ones we are using from the Brazilian Chamber of Deputies (see Sec. 3.1).

3. Method

3.1. Corpora

The Brazilian House of Representatives processes approximately 30 thousand bills[1] every year. Each bill needs to be formalized as an initial legislative document *draft* and an optional justification document. For a typical bill, a large number of documents, in different formats, is produced and submitted to the Legislative Consulting (CONLE), an advisory body of the House, whose main role is to provide the support to the law making process.

The process starts through *job requests* (legislative consultations). The *job requests* are the queries and represent the user's input to the system. While the bills and other *job requests* are the output answer, ranked according to a matching rate between the documents and the query. Thus, two legislative corpora were used to build and validate this research: the *Bills* and the *Job Request* corpora. The former is made available, while the latter has confidential information and cannot be made available[2]. Both are detailed in the following subsections.

3.1.1. Bills

For the experiments, the three most common types of bills were selected: Law Project (Projeto de Lei - PL), Complementary Law Project (Projeto de Lei Complementar - PLC), and Constitutional Amendment Proposal (Proposta de Emenda Constituicional - PEC). As a result, the final corpus had 48,555 proposals. The attribute *imgArquivoTeorPDF*, which is the bill itself, was used in the experiments. It has an average of 300 words.

[1] Legislative Information System - SiLeg

[2] https://drive.camara.leg.br/s/c3p2nLgLRcMz6eX

3.1.2. Job Request

This corpus represents the user query and contains 295 anonymized *Job Requests* from 2019. Data identifying the parliamentarian who made the *Job Request* to CONLE were removed. This corpus has two attributes: *NUMERO-PROPOSICAO-SILEG* and *TxTAssunto*. The former contains the number of the SiLeg bill that was originated from the *Job Request* specified in the latter attribute. Table 1 shows examples of parliamentarians' job requests. Most job requests have between 10 and 40 words and other files may be attached to it, such as: images, spreadsheets, links, and other documents.

Table 1. Samples from anonymized Job Request corpus.

NUMERO-PROPOSICAO-SILEG	TxTAssunto
PL XXXX/2019	Projeto para restabelecer na CLT a proibição de terceirização para atividade fim (Project to prohibit the outsourcing of core activity in the CLT)
PL XXXX/2019	Criação de PL, com base nos dois esboços encaminhados anexo. (Make of bill based on the two sketches sent in the attachment)
PL XXXX/2019	Solicito parecer pela aprovação de acordo com a solicitação XXXX/AAAA. (Request an opinion for the approval according to job request number XXXX/AAAA.)
PL XXXX/2019	Complementar parecer em função da apensação do PL XXXX/AA ao mesmo (Complementary opinion according to the PL XXXX/AA)
PL XXXX/2019	Parlamentar solicita aprovação (Parliamentarian requests approval)

3.2. Preprocessing

Both corpora presented in previous subsections had their texts converted to lowercase and the punctuation was removed. After this, every word was represented by its stem [Hotho et al. 2005], according to the evaluated algorithm. All preprocessing steps were performed using the Python NLTK library. The pipeline is available here[3]. Table 2 shows the application of four stemming algorithms used in the experiments.

- NoStem: generates no reduction of terms.
- Porter: originally written for the English language, in 1980, and adapted to the Portuguese language later. Porter is a full stemming algorithm that is based on a series of 5 conditional rules that are applied in sequence to remove the suffixes [Porter 1980]. We used the Snowball implementation available on NLTK.
- RSLP (Removedor de Sufixos da Lingua Portuguesa): a rule-based algorithm developed by [Orengo and Huyck 2001] and later improved in 2006 [Orengo et al. 2006]. Like Porter, it also applies successive steps to remove the suffixes. However, as it was developed specially for the Portuguese language, it has more rules than Porter. It has 8 steps and also presents a list of exception which prevents the algorithm from removing suffixes of words that have endings that are similar to suffixes. It was called STEMP, before.
- RSLP-S: is a variation of the RSLP algorithm, which applies only the first rule of RSLP that deals with the plural reduction. We implemented the algorithm in the Python language, based on [Orengo et al. 2006].
- Savoy (UniNE): developed by Jacques Savoy in 2006, this algorithm presents stemmers for various languages, including Portuguese. It is simpler than the others, as it has less rules. It removes inflections attached to both nouns and adjectives, based on rules for the plural and feminine form. We implemented the algorithm in the Python language, based on [Savoy 2006].

[3]https://github.com/Convenio-Camara-dos-Deputados/BM25-Experiments

Table 2. Example of stemming for each algorithm used in the experiments

NoStem	projeto	estado	solicito	deputado	aprovação	criação	federal
Porter	projet	estad	solicit	deput	aprov	criaçã	federal
RSLP (STEMP)	projet	est	solicit	deput	aprov	cri	feder
RSLP-S	projeto	estado	solicito	deputado	aprovação	criação	federal
Savoy (UniNE)	projet	estad	solicit	deputad	aprovaca	criaca	federal

3.3. Information Retrieval

Best Match 25 (BM25) [Robertson et al. 1994] is the most well-known scoring function for "bag of words" document retrieval [Kamphuis et al. 2020]. It is derived from the binary independence relevance model to include within-document term frequency information and document length normalization in the probabilistic framework for IR [Robertson and Zaragoza 2009]. The algorithm has also been used successfully in the retrieval of legal documents [N de Oliveira and C Junior 2018, Gomes and Ladeira 2020, Chalkidis et al. 2021]. We have implemented two variants presented in [Trotman et al. 2014] using the Python language.

The Okapi BM25's [Robertson et al. 1994] scoring function estimates the relevance of a document d to a query q, based on the query terms appearing in d, regardless of their proximity within d: where q_i is the *i-th* query term, with $idf(q_i)$ inverse document frequency and $tf(q_i, d)$ term frequency. BM25L [Lv and Zhai 2011] is built on the observation that the Okapi variant penalizes more longer documents compared to shorter ones. It *shifts* the term frequency normalization formula to boost scores of very long documents.

3.4. Evaluation

3.4.1. Stemming algorithms evaluation

As in [de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018], to assess the algorithm's capacity of dimensionality reduction, we considered the average number of unique terms (UDT) obtained by each stemmer and its average percentage of reduction (RP). Those measures are computed as:

- Unique Terms (UDT_s) = Frequency of unique terms after document stemming.
- Average of unique terms: $\mu = (UTD_{S1} + UTD_{S2} + +UTD_{Sn})/n$.
- Reduction percentage: $RP_R = 100 - (UTD_S \times 100)/UTD_{NoStem}$.
- Average of reduction percentage: $\mu = (RP_{S1} + RP_{S2} + ... + RP_{Sn})/n$.

3.4.2. Information retrieval evaluation

In our corpora, we have only a list of relevant documents. Therefore, we have evaluated the results in terms of *Recall* (R), which is the fraction of relevant documents that are retrieved. We analyzed the results from R@1 to R@20 (Recall at 1 document to Recall at 20 documents).

4. Results and discussion

The next subsections present and discuss the results for the stemming algorithms evaluation (Section 4.1) and for the analysis of their impact in the IR task (Section 4.2).

4.1. Stemming algorithms evaluation

Figure 1 presents the average number of unique terms per document (UTD) obtained per stemmer in the Bills corpus (1A) and in the Job Request corpus (1B). The results in both corpora were the same, in which the RSLP algorithm showed the largest reduction of unique terms: it decreased the dimensionality almost by 50, in average, when dealing with bills, and by 2.5 dealing with job requests. RSLP-S, in its turn, presented the smallest dimensionality reduction, while Porter and Savoy achieved similar UTD. This finding was the same as that one obtained by [de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018], when dealing with jurisprudential data. In their work, RSLP also showed the best capacity in terms of UTD, while RSLP-S was the worst.

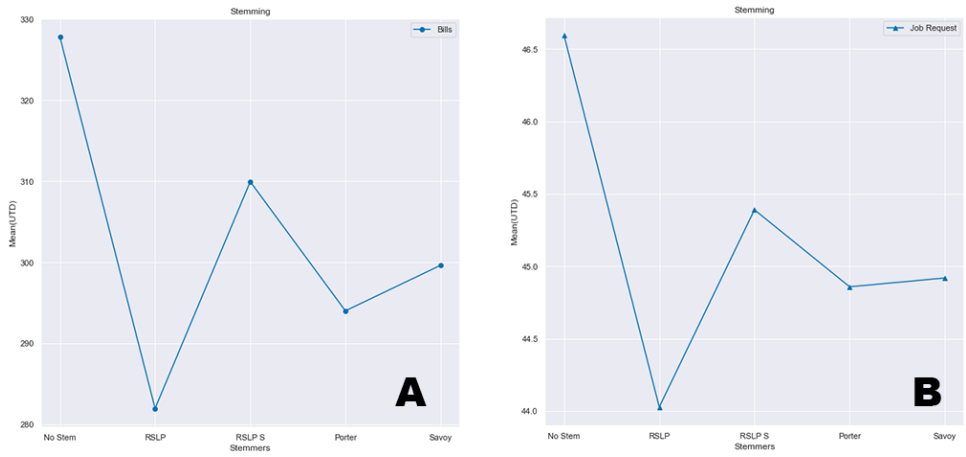


Figure 1. Average UTD per document obtained by each stemmer in the Bills corpus (A) and the Job Request corpus (B).

By the analysis of the average percentage of reduction per document (RP) in both datasets (Figure 2) we could also confirm the findings of [de Oliveira and Colaço Júnior 2017, N de Oliveira and C Junior 2018]. The RSLP algorithm achieved the best percentage of reduction as well.

The studies of [Orengo et al. 2006] and [Flores et al. 2010, Flores and Moreira 2016] did not analyze the stemming algorithms using the same metrics as we did (UTD and RP); however, in their experiments, RSLP was considered the best stemmer in terms of reduction of terms and Error Rate Relative to Truncation.

So, we can conclude that RSLP is the most effective stemmer in terms of dimensionality reduction for the legislative corpora analyzed, while RSLP-S presented the worst results. These results confirm the ones found in the literature using datasets from other domains, including jurisprudence. In addiction, it is worth mentioning that the RSLP-S performance may be explained due to how it works: by focusing only on plural reduction.

4.2. Information retrieval evaluation

Assessing the impact of stemming in the IR task, Figure 3 brings the Recall obtained by Okapi BM25 (3A) and BM25L (3B) using each stemmer for the Bills corpus.

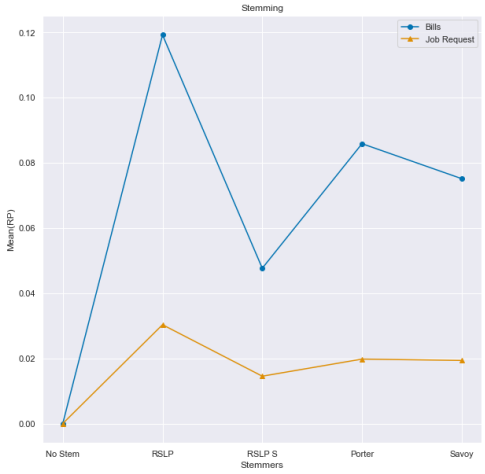


Figure 2. Average RP per document obtained by each stemmer.

According to the Recall@10 results, we could observe that using Okapi BM25 without radicalization (NoStem) achieved better results than using stemming algorithms. This confirms the finding of [N de Oliveira and C Junior 2018], that the use of stemmers deteriorate the original BM25 performance. However, the same was not observed using BM25L variant, for which stemming improved the results. For this IR algorithm, Savoy was the best stemmer in terms of Recall@10. Nevertheless, considering the Recall@20 measure, the Savoy algorithm outperformed the use of IR without radicalization, using the Okapi BM25. While for the use of BM25L, Savoy and RSLP-S were the best algorithms.

We also performed a statistical analysis using the Friedman test [Friedman 1937] and the Nemenyi post-hoc test [Nemenyi 1963], considering the values of Recall@1 to Recall@20 for each stemming algorithm. The Friedman test pointed out that there was a difference between the algorithms for both Okapi BM25 and BM25L, while the Nemenyi post-hoc test indicated which algorithms showed a difference. As we can see in the CD diagrams from Figure 4, for Okapi BM25, IR without stemming (NoStem) achieved the best results, being statistically similar only to Savoy and statistically better than the others. However, for BM25L, the use of Savoy was the best and statistically better than NoStem.

In this sense, we could notice that the adoption of radicalization for legislative documents retrieval depends on the IR algorithm chosen: using Okapi BM25, it is recommended that no stemmer is used; while using BM25L, the dimensionality reduction may improve the IR performance. Meanwhile, analyzing just the performance of the different stemmers, Savoy and RSLP-S can be pointed out as the best ones in terms of Recall for the scenarios analyzed; while RSLP and Porter (Snowball) achieved very poor results.

Finally, comparing these findings with the analysis from Section 4.1, we can state that a great dimensionality reduction does not indicate a better performance for IR. The RSLP algorithm showed the best results in terms of reduction, but when it was used for IR, it deteriorated the BM25 performances. On the other hand, RSLP-S was the worst in terms of UTD and RP, while one of the best for documents retrieval.

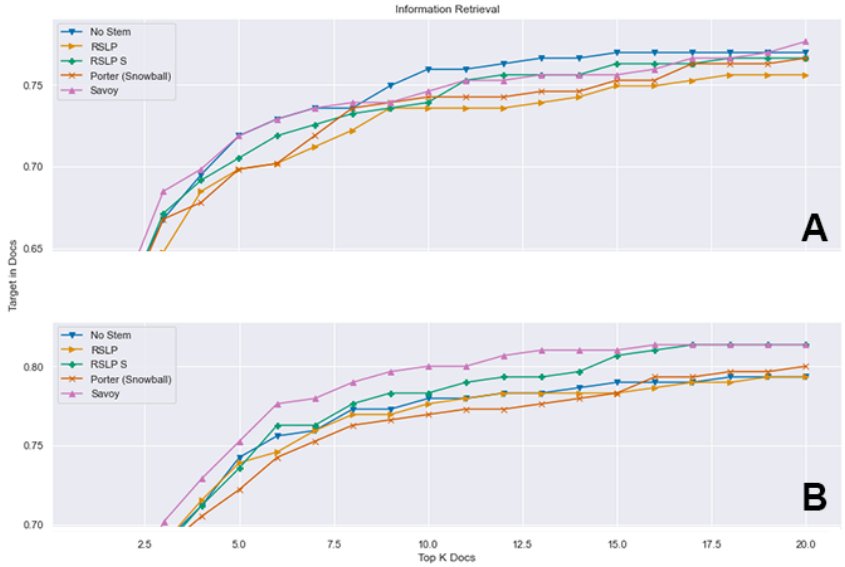


Figure 3. Recall achieved by Okapi BM25 (A) and BM25L (B) using each stemmer.

Figure 4. Results of Nemenyi post-hoc test for Okapi BM25 (A) and BM25L (B).

5. Conclusion

This paper presents a contribution related to the application of stemming algorithms on the legislative domain for terms of data dimensionality reduction and evaluates the efficiency of algorithms in the IR task. Four Portuguese stemmers were evaluated: Porter, RSLP (STEMP), RSLP-S, and Savoy (UniNE). The average number of unique terms per document and the average percentage of reduction per document were used to evaluate the stemmers. For the IR task, two BM25 variants were evaluated using the Recall measure.

The RSLP algorithm showed the largest reduction of unique terms (UDT), while RSLP-S presented the smallest dimensionality reduction. The RSLP also achieved the best percentage of reduction (RP). Assessing the impact of stemming in the IR task with the BM25L, Savoy was the best stemmer in terms of Recall@10, while, using Recall@20, Savoy and RSLP-S achieved the same result. For the Okapi BM25, with Recall@10, we could observe that IR without radicalization (NoStem) achieved better results than using stemming algorithms, confirming the finding of [N de Oliveira and C Junior 2018].

We conclude that the adoption of radicalization for legislative documents retrieval depends on the IR algorithm chosen and that a great dimensionality reduction does not indicate a better performance for IR. For future work, we intend to analyze the impact of the reduction using other corpora in the Legislative domain, measuring its impact on IR.

References

Almeida, P. G. R. (2021). Uma jornada para um Parlamento inteligente: Câmara dos Deputados do Brasil. *Red Información*, 24.

Alvares, R. V., Garcia, A. C. B., and Ferraz, I. (2005). Stembr: A stemming algorithm for the brazilian portuguese language. In Bento, C., Cardoso, A., and Dias, G., editors, *Progress in Artificial Intelligence*, pages 693–701, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chalkidis, I., Fergadiotis, M., Manginas, N., Katakalou, E., and Malakasiotis, P. (2021). Regulatory compliance through Doc2Doc information retrieval: A case study in EU/UK legislation where text similarity has limitations. *arXiv preprint arXiv:2101.10726*.

de Oliveira, R. A. and Colaço Júnior, M. (2017). Assessing the impact of stemming algorithms applied to judicial jurisprudence-an experimental analysis. In *International Conference on Enterprise Information Systems*, volume 2, pages 99–105. SCITEPRESS.

Flores, F. N. and Moreira, V. P. (2016). Assessing the impact of stemming accuracy on information retrieval–a multilingual perspective. *Information Processing & Management*, 52(5):840–854.

Flores, F. N., Moreira, V. P., and Heuser, C. A. (2010). Assessing the impact of stemming accuracy on information retrieval. In *International Conference on Computational Processing of the Portuguese Language*, pages 11–20. Springer.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

Gomes, T. and Ladeira, M. (2020). A new conceptual framework for enhancing legal information retrieval at the brazilian superior court of justice. In *Proceedings of the 12th International Conference on Management of Digital EcoSystems*, page 26–29.

Hotho, A., Nürnberger, A., and Paaß, G. (2005). A Brief Survey of Text Mining. *Journal for Computational Linguistics and Language Technology*, pages 1–37.

Kamphuis, C., de Vries, A. P., Boytsov, L., and Lin, J. (2020). Which BM25 do you mean? A large-scale reproducibility study of scoring variants. In *Advances in Information Retrieval*, pages 28–34.

Lv, Y. and Zhai, C. (2011). When documents are very long, BM25 fails! In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1103–1104.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, USA.

Maxwell, K. T. and Schafer, B. (2008). Concept and context in legal information retrieval. *Frontiers in Artificial Intelligence and Applications*, 189:63–72.

Moral, C., de Antonio, A., Imbert, R., and Ramírez, J. (2014). A Survey of Stemming Algorithms in Information Retrieval. *Information Research: An International Electronic Journal*, 19(1)(n1):22.

N de Oliveira, R. A. and C Junior, M. (2018). Experimental analysis of stemming on jurisprudential documents retrieval. *Information*, 9(2):28.

Nemenyi, P. (1963). *Distribution-free multiple comparisons*. PhD thesis, Princeton University.

Orengo, V. M., Buriol, L. S., and Coelho, A. R. (2006). A study on the use of stemming for monolingual ad-hoc portuguese information retrieval. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 91–98. Springer.

Orengo, V. M. and Huyck, C. R. (2001). A stemming algorithmm for the portuguese language. In *SPIRE*, volume 8, pages 186–193.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, 40:211–218.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1994). Okapi at TREC-3. In *TREC*.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Savoy, J. (2006). Light stemming approaches for the french, portuguese, german and hungarian languages. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1031–1035, New York, NY, USA. Association for Computing Machinery.

Trotman, A., Puurula, A., and Burgess, B. (2014). Improvements to BM25 and language models examined. *ACM International Conference Proceeding Series*, 27-28-Nove:58–65.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

verBERT: Automating Brazilian Case Law Document Multi-label Categorization Using *BERT*

Felipe R. Serras[1], **Marcelo Finger**[1]

[1]Institute of Mathematics and Statistics (IME) – University of São Paulo (USP)
R. do Matão, 1010 – Butantã – São Paulo – SP – Brazil – 05508-090

{frserras,mfinger}@ime.usp.br

Abstract. *In this work, we carried out a study about the use of attention-based algorithms to automate the categorization of Brazilian case law documents. We used data from the Kollemata Project to produce two distinct datasets with adequate class systems. Then, we implemented a multi-class and multi-label version of BERT and fine-tuned different BERT models with the produced datasets. We evaluated several metrics, adopting the micro-averaged F1-Score as our main metric for which we obtained a performance value of* $\langle \mathcal{F}_1 \rangle_{micro} = 0.72$ *corresponding to gains of 30 percent points over the tested statistical baseline.*

1. Introduction

In this work, we explore the use of *BERT* [Devlin et al. 2019] to develop a prototype to automate the categorization of Brazilian case law documents, known as "*verbetação*". This categorization has been done manually for the past 25 years by experts in the field of law, without taking advantage of knowledge engineering techniques; as a result, the categories are not organized as any computationally coherent data structure. Its automation could, therefore, save time and labor, in addition to optimizing its ability to represent knowledge. The problem was brought to us by the *Kollemata* Project team, which organized a library of Brazilian jurisprudence, containing more than 24,000 case law documents from the property law field, and whose dataset was used as the basis for this research.

Unlike other problems in NLP, this one was not clearly associated with a predefined task. We raised several modeling possibilities and chose multi-label categorization because of the high value of having an unified set of predetermined thematic categories for this task, manifested both in the literature of the field and by law professionals we consulted.

Considering the scarce literature on this problem, especially in Brazilian Portuguese, our goals were (i) to provide an initial proposal for modeling the problem, compatible with the manifested needs of the consulted professionals, (ii) to assess the potential performance of the main state-of-the-art methods (in this case BERT) on this modeling, and (iii) to map the main obstacles to performance improvement on the adopted modeling. The results of our work are prototypical and seek, mainly, to start the debate and indicate the future paths that research on this topic could take.

To achieve these goals, we reformulated the *Kollemata* dataset with an appropriate class system, and implemented a set of *BERT* classifiers, which can serve as prototypes to automate the categorization of Brazilian case law documents, deepening our understand of the addressed problem and the used algorithm and methodology.

2. Related Work

Following the proposition of *BERT*, which included the presentation of the *Multilingual BERT* model, the *BERTimbau* model [Souza et al. 2020] - the first *BERT* model trained entirely in Brazilian Portuguese - was developed and used in the automation of several tasks in Brazilian Portuguese [Leite et al. 2020, Souza et al. 2019, Salvatore et al. 2019].

As far as we know, this is the first work to explore the use of *BERT* to automate case law documents categorization in Brazilian Portuguese. Other works explore the same problem in other languages or dialects of Portuguese [Calambás et al. 2015, Gonçalves and Quaresma 2003, Gonçalves and Quaresma 2005, Feinerer and Hornik 2008, Mencía and Fürnkranz 2010], and one work explores a single-label version of the same problem with less data in Brazilian Portuguese [de Colla Furquim and De Lima 2012]. All of these articles studied algorithms created before *BERT*.

Regarding our efforts to reorganize the data class system, we used clustering techniques to produce an organized class hierarchy, which is not a new ideia for general [Aggarwal and Zhai 2012] or legal applications [Moens 2001]. However, here we adopted a methodology designed to meet the specific needs of this task.

Only few recent works relate to ours, both in method and type of task, as the articles of [Villata et al. 2020] and [Chalkidis et al. 2019]. In the first one, the authors explore different applications of machine learning to Italian law documents, including the use of the *Multilingual BERT* model to perform the categorization of case law documents under a proprietary taxonomy. In the second one, several algorithms, including BERT, were tested to generate a multi-label classifier for European legislative documents in English, based on a large system of topics. These two works are those whose results are most comparable to ours.

3. Problem

In Brazil, case law documents are called "*acórdãos*" and are used to register the decision of a court and to serve as reference to similar cases in the future (jurisprudence). Searching for these documents is a recurrent task for law professionals and to facilitate it, all case law documents contain a component of jurisprudence ("*ementa jurisprudencial*"), formed by (i) a brief summary of the documents content and (ii) a header, containing a set of descriptive terms that represent the main legal topics covered in the decision. In Portuguese, these terms are called "*verbetes*" and the header is called "*verbetação*". Figure 1 contains a gold standard example of a Brazilian component of jurisprudence [Guimarães 2004].

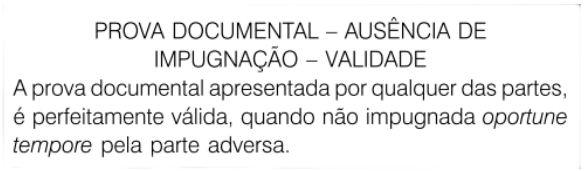

PROVA DOCUMENTAL – AUSÊNCIA DE IMPUGNAÇÃO – VALIDADE
A prova documental apresentada por qualquer das partes, é perfeitamente válida, quando não impugnada *oportune tempore* pela parte adversa.

Figure 1. A gold standard example of a component of jurisprudence from a Brazilian case law document, extracted from [Guimarães 2004].

The purpose of this component of jurisprudence is to help readers in assessing whether a document is relevant or not to them. However, [Guimarães 2004] identifies

problems in the ways these headers are produced, including redundancy, usage of generic terms, lack of a consistent vocabulary, and lack of well-defined syntactic structures, both within the terms and between them. These problems affect the appropriate usage of the component of jurisprudence and get in the way of having a well formed and nationally validated set of case law descriptor terms, seen as an important asset within the Brazilian theory of law.

As stated in Section 1, given the importance of having this set of predetermined descriptor terms, we decided to model it as a task of multi-class and multi-label text categorization i.e. the task of assigning to each document d_j in a set of natural language documents $\mathcal{D}$, a variable number of categories c_i, taken from a set of multiple possible predefined categories $\mathcal{C}$ [Sebastiani 2002]. Also, fitting the data to this model could serve as basis for the resolution of several problems identified by [Guimarães 2004].

4. Materials and Methodology

4.1. *Kollemata* and *verBERT*

We used data from the *Kollemata* Project[1]. Its database contains more than 24,000 entries, each one corresponding to one case law document from the property law field and containing its summary and header[2].

Our goal was to use the pairs of summary and header content as the examples to BERT fine-tuning, aiming to automatically generate the terms of the header from the summary content. The procedures performed to prepare this data for this usage are explained in Section 5.

We implemented a *Python3* program called *verBERT*, that uses the Hugging Face Transformers Package [Wolf et al. 2019] to fine-tune, evaluate and test *BERT* multi-label classifiers using pre-treined *BERT* models. In doing so, we refactored, adapted and incremented the examples of Hugging Face for generating BERT Classifiers over GLUE [Wang et al. 2019], with copyright from Hugging Face, Google and Nvidia, but available under the Apache 2.0 License[3].

Our multi-label categorization architecture was inspired by [Trivedi 2019] and uses a simple feed-forward network coupled with sigmoid functions σ as the categorization layer (Equations 1 and 2, where $\mathbf{a}$ is the *BERT* output representation, $W^C : W^C \in \mathbb{R}^{d \times |\mathcal{C}|}$ a linear transformation matrix, $\mathbf{b}$ a bias vector, and d is the characteristic dimension of the model) [Han and Moraga 1995], and the mean of the binary cross-entropy [Goodfellow et al. 2016, p. 65-66] over the labels as the loss function to be minimized using Adam optimizer [Kalchbrenner et al. 2014, Goodfellow et al. 2016, p. 305-306] with weight decay regularization [Loshchilov and Hutter 2017].

$$\mathbf{c} = \varsigma(\mathbf{a}W^C + \mathbf{b}) \tag{1}$$

[1] `https://www.kollemata.com.br/`

[2] The *Kollemata* Project dataset was given to our research by the *Kollemata* Project team. The data set will not be publicly available for now, as Brazilian authorities are reviewing the correct way to make these documents available on a large scale without exposing sensitive data.

[3] The *verBERT* source code, as well as other resources, are available in our repository: `https://github.com/frserras/verbert-categorization`

$$\varsigma(\mathbf{x} = \{x_1, ..., x_n\}) = \{\sigma(x_1), ..., \sigma(x_n)\} \quad (2)$$

We evaluated and compared three pre-trained *BERT* models as bases for our classifier generation: *Multilingual BERT*, a base model trained over 104 languages on Wikipedia data [Devlin et al. 2019], and the two variants (base and large) of the *BERTimbau* Model, pre-trained entirely in Portuguese [Souza et al. 2020, Souza et al. 2019].

4.2. Metrics and Baseline

To compare the models produced by *verBERT*, we adopted three performance metrics commonly used in problems of categorization and information retrieval: precision $\mathcal{P}$, recall $\mathcal{R}$, and F1-Score $\mathcal{F}_1$ [Godbole and Sarawagi 2004]. For each metric $\mathcal{E}$ we computed its three averaged multi-label versions: the micro-average $\langle\mathcal{E}\rangle_{micro}$, the macro-average $\langle\mathcal{E}\rangle_{macro}$ and the average by instance $\langle\mathcal{E}\rangle_{inst}$ [Koyejo et al. 2015]

We also used the accuracy metric [Godbole and Sarawagi 2004], for which we selected two versions for the multi-label case: the Hamming accuracy $\mathcal{A}$ i.e. the percentage of labels correctly predicted over all examples, and the subset accuracy $\dot{\mathcal{A}}$, the percentage of examples for which all labels were correctly predicted.

We selected the $\langle\mathcal{F}_1\rangle_{micro}$ as our main metric, since it tends to be more appropriate for information retrieval cases with strong unbalanced categories, like ours. As secondary metrics, we adopted (i) the $\langle\mathcal{F}_1\rangle_{macro}$, which shows us the impact of category imbalance on performance, if compared with its micro-averaged counterpart, and (ii) the subset accuracy $\dot{\mathcal{A}}$, because is a more robust metric for problems of categorization, that serves as a lower bound to our performance.

Besides the models produced by *verBERT*, we implemented a simple statistical baseline method, where the $n = 5$ most common categories of the dataset were used to categorize all of its entries[4]. Its results are presented in Table 2. This type of baseline, despite its simplicity, is a robust option for unbalanced category systems, where betting on the most likely categories can be a good strategy for blindly maximizing performance.

5. Data preparation and Ontological Adjustment

We used the pairs [summary, header], from the *Kollemata* Project dataset and preprocessed them by (i) filtering corrupted *HTML* notation from the summaries, (ii) unifying the terms for law entities across the dataset, and (iii) mapping and unifying different segmentation symbols between descriptor terms. The data was structured in a list of entries, each one corresponding to a case law document and containing the summary and an ordered list of its header descriptor terms.

We then performed an exploratory analysis on the dataset, to identify its main properties and eventual obstacles for automation. We evaluated the distributions, across the data, of summary size, number of categories per header, presence of the descriptor terms in their respective summary, and category occurrence.

From the summary size distribution, we observed that the expected number of subwords by summary is less than the internal BERT dimensionality d for our three models,

[4]We tested different values for n and choose the one that maximized the baseline performance for our main metrics.

which confirms that this configuration is in the range where BERT is expected to be more efficient than recurrent models [Devlin et al. 2019].

The categories per header distribution revealed an average of 5 descriptor terms per document, while the distribution of presence of the descriptor terms in their respective summary revealed that most summaries contain a fraction of their respective descriptor terms, but not all. Both of these results characterize a scenario compatible with modeling the problem as a multi-label classification task.

On the other hand, the category occurrence distribution revealed a total number of categories (20,780) close to the number of documents in the dataset, most of them with low occurrence. This is due to the high specificity of the descriptor terms, that contain many concepts and sub-concepts within themselves. The descriptor terms "*ineficácia da adjudiação*", "*ineficácia da alienação*" and "*ineficácia da caução*" are a good example. All of them represent, at the same time, the concept of "*ineficácia*" and another sub-concept, unique to each of these terms. Ideally, the concept and sub-concept would be separated somehow.

This corroborates the problems pointed out by [Guimarães 2004], highlighted in Section 3. To enable the treatment of the problem via multi-label categorization, we performed an ontological adjustment step, in an attempt to extract the concepts contained in each descriptor term and organize them in a class system with reasonable size.

Before that, however, we also examined the correlation between the orders of appearance of any two terms in the headers, expecting to detect any type of underlying hierarchy between them that could be used in our ontological adjustment. We observed a strong pairwise correlation, but not to the point of building longer and more complex hierarchical lineages between the categories. From this we concluded that segmenting the categories in their components and then build a hierarchical structure from them, taking advantage of these co-occurrences was the best way to adequate the class system.

During this ontological adjustment, category descriptor terms were tokenized in its component words, stop words were removed using the NLTK[5] Portuguese stop words list, and then stemmed with a manually enriched version of the RSLP Stemmer [Huyck and Orengo 2001]. We removed the terms with less then 5 occurrences to eliminate spurious terms and then assigned paternity relations between terms with high co-occurrence.

The terms in the top of the obtained hierarchy with low occurrence were grouped under the "Others" category, while the ones with high occurrence were automatically clustered in 25 super-categories, each connected to the component terms of the original categories by the hierarchical structure. We clustered the categories using a K-means clusterizer applied to arrays with identifiers of the documents in which each category appeared[6] [Manning and Schütze 1999, p. 515-518]. These arrays had their dimension previously reduced to $d = 50$ using the *scikit-learn* Trucated SVD Algorithm[7]. We tested

[5] `https://www.nltk.org/`

[6] This clustering was performed as a pre-processing step. The classifier performs the categorization on already clustered data and therefore has no access to the arrays with identifiers of the documents in which each category appeared, which could introduce a bias towards the actual labels.

[7] `https://scikit-learn.org/stable/modules/generated/sklearn.`

different values of reduced dimension d and different clustering methods. Each combination resulted in different amounts of clusters, and the chosen combination was the one whose clustering groups were the most cohesive.

Despite being justified by the properties of the data, the ontological adjustment carried out here is extensive. For this reason, the performance values obtained should be understood as the joint performance between the categorization method and the adopted data reorganization methodology.

6. Experiments and Results

To evaluate and compare the performance of the models produced by *verBERT* and our hierarchy of categories, we executed a series of experiments. We used three pre-trained models, *Multilingual BERT*, *BERTimbau* base, and *BERTimbau* large; and two different versions of the produced hierarchy and dataset: dataset 1, produced with a smaller grouping rate and maintaining the "Others" category, and dataset 2, a slightly more homogeneous option, produced with higher grouping rate and excluding the "Others" category.

Each combination of model and dataset prompted a group of experiments, in which we varied the maximum learning rate $\lceil \eta \rceil$, reached after 50 linear warming-up steps ($2 \cdot 10^{-5}$, $4 \cdot 10^{-5}$, $5 \cdot 10^{-5}$, $1 \cdot 10^{-4}$, $1 \cdot 10^{-3}$, $5 \cdot 10^{-4}$, $5 \cdot 10^{-3}$, and $1 \cdot 10^{-2}$)[8], the maximum input sentence size $|\mathcal{S}|$ (52, 68, 131, and 200 tokens) and the categorization threshold probability P_{ct}, above which a category is assigned to a document by the model (0.25, 0.50 and 0.75).

We divided each dataset in a training set (72%), a validation set (8%) and a test set (20%). In each experiment the model was fine-tuned with the training set for 10 epochs and evaluated at regular intervals against the validation set. The intermediate model with the best performance on the validation set was then evaluated on the test set.

For that, we used two machines: (i) one with 48 cores, 378 GB of primary memory and a *NVIDIA Tesla K40c GPU* (11.4 GB), and the other (ii) with 40 cores, 62.8 GB of primary memory, and a *NVIDIA GeForce GTX1080Ti GPU* (11.7 GB). Since both machines had secondary memory limitations, *verBERT* was planned to use little storage as possible.

Comparing the results of the experiments, we observe an improvement in performance with higher values of maximum learning rate $\lceil \eta \rceil$ and input sentence size $|\mathcal{S}|$.The increase in performance with input sentence size is observed for all values of $|\mathcal{S}|$, but the increase rate decreases progressively, indicating saturation of this trend. On the other hand, the growth in performance with maximum learning rate $\lceil \eta \rceil$ is observed only up to $\lceil \eta \rceil = 5 \cdot 10^{-4}$, from which point on, performance rates fall sharply. Our investigations suggest that this is due to a local minimum, where the optimizer got stuck.

We also observed that the categorization threshold probability dictates the optimization strategy adopted by the algorithm: with higher values of P_{ct}, it is easier to maximize precision first and then slightly decrease it to increase recall and F1-score. In

`decomposition.TruncatedSVD.html`

[8]We used learning rate values based on the ones used in [Devlin et al. 2019] and [Souza et al. 2019], and included larger values to get a more general picture of the performance evolution.

Table 1. Best results for the main metrics and respective parameters from each group of experiments on the test set.

↓Dataset/Model→	**BERTimbau LARGE**	
Dataset 1	$\langle \mathcal{F}_1 \rangle_{micro} = 0.71$	$\dot{\mathcal{A}} = 0.24$
	$\langle \mathcal{F}_1 \rangle_{macro} = 0.66$	$(P_{limiar} = 0.25; \lceil \eta \rceil = 1 \cdot 10^{-4}; \lvert \mathcal{S} \rvert = 200)$
Dataset 2	$\langle \mathcal{F}_1 \rangle_{micro} = 0.72$	$\dot{\mathcal{A}} = 0.38$
	$\langle \mathcal{F}_1 \rangle_{macro} = 0.71$	$(P_{limiar} = 0.50; \lceil \eta \rceil = 1 \cdot 10^{-4}; \lvert \mathcal{S} \rvert = 131)$
↓Dataset/Model→	**Multilingual BASE**	
Dataset 1	$\langle \mathcal{F}_1 \rangle_{micro} = 0.69$	$\dot{\mathcal{A}} = 0.24$
	$\langle \mathcal{F}_1 \rangle_{macro} = 0.59$	$(P_{limiar} = 0.50; \lceil \eta \rceil = 1 \cdot 10^{-4}; \lvert \mathcal{S} \rvert = 200)$
Dataset 2	$\langle \mathcal{F}_1 \rangle_{micro} = 0.71$	$\dot{\mathcal{A}} = 0.37$
	$\langle \mathcal{F}_1 \rangle_{macro} = 0.70$	$(P_{limiar} = 0.50; \lceil \eta \rceil = 5 \cdot 10^{-5}; \lvert \mathcal{S} \rvert = 200)$
↓Dataset/Model→	**BERTimbau BASE**	
Dataset 1	$\langle \mathcal{F}_1 \rangle_{micro} = 0.70$	$\dot{\mathcal{A}} = 0.24$
	$\langle \mathcal{F}_1 \rangle_{macro} = 0.59$	$(P_{limiar} = 0.50; \lceil \eta \rceil = 1 \cdot 10^{-4}; \lvert \mathcal{S} \rvert = 200)$
Dataset 2	$\langle \mathcal{F}_1 \rangle_{micro} = 0.72$	$\dot{\mathcal{A}} = 0.38$
	$\langle \mathcal{F}_1 \rangle_{macro} = 0.70$	$(P_{limiar} = 0.25; \lceil \eta \rceil = 4 \cdot 10^{-5}; \lvert \mathcal{S} \rvert = 200)$

contrast, with smaller threshold probability values, it is easier to maximize recall first and then decrease it optimally, to enhance precision and F1-score.

To provide a global view, Table 1 presents the main performance metrics in the best case for each group of experiments, and the parameters for which these results were obtained.

We observed better results on average for the large model, than for the two base models, for both datasets. However, this improvement is low: only two percent point increase in $\langle \mathcal{F}_1 \rangle_{micro}$ for the best case. This seems to indicate the superiority of the larger model and the robustness of the smaller ones at the same time. This improvement is larger for macro-F1 over the dataset 1. Since the dataset 1 is less homogeneous and the macro-F1 is a metric more sensitive to the lack of homogeneity, this suggests that the large model is better at dealing with less homogeneous category systems.

Comparing the results for the *BERTimbau* models with the results for the *Multilingual* model, we see that using Portuguese based models improves average performance, but again, this improvement is low, what may indicate the robustness of the *Multilingual* model.

Looking at the impact of data homogeneity over performance, we see that the use of dataset 2 instead of dataset 1 brings an improvement to all models, specially in the subset accuracy $\dot{\mathcal{A}}$, which indicates a significant increase in our performance lower bound. Furthermore, we also see an increase in the macro-averaged F1, which reveals that a small improvement in homogeneity was enough to cause a great impact in the gap between the micro-averaged and the macro-averaged F1 scores. These results should nonetheless be interpreted with caution, given the difference between the test sets of dataset 1 and dataset 2 . Also, with dataset 2 we obtained our best model, the $verBERT\star$, obtained by fine-tuning *BERTimbau* large with $P_{ct} = 0.5$, $\lceil \eta \rceil = 1 \cdot 10^{-4}$ and $\lvert \mathcal{S} \rvert = 131$. In Table 2 we compare all the performance metrics obtained by $verBERT\star$ and by our baseline, described in Section 4

Table 2. A comparison of the performance results obtained by the $verBERT\star$ model and by our statistical baseline, for all performance metrics.

	$\langle\mathcal{P}\rangle_{micro}$	$\langle\mathcal{P}\rangle_{macro}$	$\langle\mathcal{P}\rangle_{inst}$	$\langle\mathcal{R}\rangle_{micro}$	$\langle\mathcal{R}\rangle_{macro}$	$\langle\mathcal{R}\rangle_{inst}$
Baseline	0.33	0.06	0.33	0.47	0.19	0.49
$verBERT\star$	0.77	0.77	0.79	0.67	0.66	0.73
	$\langle\mathcal{F}_1\rangle_{micro}$	$\langle\mathcal{F}_1\rangle_{macro}$	$\langle\mathcal{F}_1\rangle_{inst}$	$\mathcal{A}$	$\dot{\mathcal{A}}$	
Baseline	0.39	0.09	0.37	0.80	0.00	
$verBERT\star$	0.72	0.71	0.73	0.95	0.38	

7. Conclusions

The $verBERT\star$ model achieves results significantly superior to the baseline for all metrics. Beyond that, the value obtained by $verBERT\star$ for our main metric, the micro-averaged F1 score, is close to the 0.732 value obtained by [Chalkidis et al. 2019], and is higher then the 0.505 value, obtained by [Villata et al. 2020] [9].

These results are quite positive given the difficulties in modeling the problem, and indicate that our approach is promising. However, (i) a direct comparison would be inappropriate, given the methodological differences between our work and those cited, (ii) the observed performance is still far from the optimal values obtained for other applications in NLP, and (iii) to obtain such a level of performance it was necessary to carry out a heavy pre-processing in which we observed several of the taxonomic problems of the generation of headers, pointed out in the literature. This indicates that the lack of standardization in headers is the main obstacle to achieving better performance results.

Despite the usefulness of our models as categorization prototypes that could already be used to assist human classifiers, we believe that, given these results, better performance values could only be obtained (i) by exploring models other than the categorization, which would compromise the taxonomic value of the descriptor terms pointed by field specialists, or (ii) by revisiting the production system of the headers in its core, in order to manually generate a system of categories that is appropriate for the main activity and that could be used in the automation process without so much pre-processing . In addition, (iii) repetitions of our experiments could bring greater statistical value to these results, mitigating the uncertainty brought by the differences between the class systems of our two datasets, and (iv) the exploration of more robust baselines based on other natural language processing techniques could help us better understand the quality of the performance obtained by our method. These are the main issues we would like to address in future works.

Acknowledgements

This work was carried out as a master's research project [Serras 2021] at the Center for Artificial Intelligence (C4AI-USP), with support by the São Paulo Research Foundation (FAPESP grant #2019/07665-4) and by the IBM Corporation. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. We would like to thank Dr. Sérgio Jacomino, his team from the *Kollemata Project*, Professors Adriana Unger, Juliano Maranhão, Renata Vieira, Denis Mauá and the anonymous reviwers for their help.

[9] As mentioned in Section 2, these two works are those whose methods, applications, and therefore results, are most comparable to ours.

References

Aggarwal, C. C. and Zhai, C. (2012). A survey of text clustering algorithms. In *Mining text data*, pages 77–128. Springer.

Calambás, M. A., Ordóñez, A., Chacón, A., and Ordoñez, H. (2015). Judicial precedents search supported by natural language processing and clustering. In *2015 10th Computing Colombian Conference (10CCC)*, pages 372–377. IEEE.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., and Androutsopoulos, I. (2019). Large-scale multi-label text classification on eu legislation. *arXiv preprint arXiv:1906.02192*.

de Colla Furquim, L. O. and De Lima, V. L. S. (2012). Clustering and categorization of brazilian portuguese legal documents. In *International Conference on Computational Processing of the Portuguese Language*, pages 272–283. Springer.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Feinerer, I. and Hornik, K. (2008). Text mining of supreme administrative court jurisdictions. In *Data Analysis, Machine Learning and Applications*, pages 569–576. Springer.

Godbole, S. and Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 22–30. Springer.

Gonçalves, T. and Quaresma, P. (2003). A preliminary approach to the multilabel classification problem of portuguese juridical documents. In *Portuguese Conference on Artificial Intelligence*, pages 435–444. Springer.

Gonçalves, T. and Quaresma, P. (2005). Is linguistic information relevant for the classification of legal texts? In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 168–176.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Guimarães, J. A. C. (2004). *ELABORAÇÃO DE EMENTAS JURISPRUDENCIAIS: elementos teórico-metodológicos*, volume 9 of *Série Monografias do CEJ*.

Han, J. and Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*, pages 195–201. Springer.

Huyck, V. and Orengo, V. (2001). A stemming algorithm for the portuguese language. In *SPIRE*, volume 1, pages 186–193.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*.

Koyejo, O., Natarajan, N., Ravikumar, P., and Dhillon, I. S. (2015). Consistent multilabel classification. In *Advances in Neural Information Processing Systems*.

Leite, J. A., Silva, D. F., Bontcheva, K., and Scarton, C. (2020). Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis. *arXiv preprint arXiv:2010.04543*.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Mit Press. MIT Press.

Mencía, E. L. and Fürnkranz, J. (2010). Efficient multilabel classification algorithms for large-scale problems in the legal domain. In *Semantic Processing of Legal Texts*, pages 192–215. Springer.

Moens, M.-F. (2001). Innovative techniques for legal text retrieval. *Artificial Intelligence and Law*, 9(1):29–57.

Salvatore, F., Finger, M., and Hirata Jr, R. (2019). A logical-based corpus for cross-lingual evaluation. *arXiv preprint arXiv:1905.05704*.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47.

Serras, F. R. (2021). Algoritmos baseados em atenção neural para a automação da classificação multirrótulo de acórdãos jurídicos. Master's thesis, Instituto de Matemática e Estatística (IME).

Souza, F., Nogueira, R., and Lotufo, R. (2019). Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*.

Trivedi, K. (2019). Multi-label text classification using bert – the mighty transformer. `https://medium.com/huggingface/multi-label-text-classification-using-bert-the-mighty-transformer-69714fa3fb3d`. Acessado em 02/03/2020.

Villata, S. et al. (2020). Natural language processing applications in case-law text publishing. In *Legal Knowledge and Information Systems: JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020*, volume 334, page 154. IOS Press.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Annotation Difficulties in Natural Language Inference

Aikaterini-Lida Kalouli[1], Livy Real[2], Annebeth Buis[3], Martha Palmer[3], Valeria de Paiva[4]

[1]Center for Information and Language Processing – LMU
Munich – Germany

[2]americanas s.a. d.lab – São Paulo, Brazil

[3]Department of Linguistics – University of Colorado at Boulder
Boulder, Colorado – USA

[4]Topos Institute
Berkeley, California – USA

kalouli@cis.lmu.de, livyreal@gmail.com

Abstract. *State-of-the-art models have obtained high accuracy on mainstream Natural Language Inference (NLI) datasets. However, recent research has suggested that the task is far from solved. Current models struggle to generalize and fail to consider the inherent human disagreements in tasks such as NLI. In this work, we conduct an experiment based on a small subset of the NLI corpora such as SNLI and SICK. It reveals that some inference cases are inherently harder to annotate than others, although good-quality guidelines can reduce this difficulty to some extent. We propose adding a Difficulty Score to NLI datasets, to capture the human difficulty level of agreement.*

1. Introduction

Natural Language Inference (NLI), the task of determining whether a premise (P) entails, contradicts or is neutral to a hypothesis (H), has recently seen tremendous progress. The increasing availability of huge datasets has facilitated the training of massive models, pushing the state-of-the-art (SOTA) to high levels of accuracy, with some papers even claiming to reach human performance [Liu et al. 2019, Zhang et al. 2019]. With these results, one might consider NLI a solved task. To date, two different strands of research attempt to show that this seemingly perfect performance does not make the task solved. One strand focuses on detecting bias or artifacts in the training sets [Gururangan et al. 2018, Poliak et al. 2018] and creating challenging datasets that expose the generalization difficulties of the models [Glockner et al. 2018, Nie et al. 2018, McCoy et al. 2019, inter alia]. Another strand of research shows that due to inherent disagreements in tasks such as NLI, current SOTA models cannot claim to capture human level inference capabilities [de Marneffe et al. 2018, Palomaki et al. 2018, Pavlick and Kwiatkowski 2019].

This paper addresses both these directions. Specifically, in this work we discuss an experiment realized at X University, which investigates whether some NLI pairs are inherently more difficult and controversial to annotate than others, leading to significantly lower Inter-Annotator-Agreement (IAA). Parallel to that, we investigate how such difficult pairs can be detected, as well as to what extent different annotation guidelines might be able to solve some of the inherent complexity. For the latter investigation, we build

on previous work by [Kalouli et al. 2017] and [Kalouli et al. 2019], who list difficult phenomena that become sources of disagreement and also propose improved guidelines for the task. Thus, in this work, our contributions are three-fold. First, we quantitatively show that both corpora under investigation contain inherently controversial pairs that lead to significantly higher disagreements than other pairs. Secondly, we propose the augmentation of the NLI annotation task with a *Difficulty Score*. Such a score can contribute to a better training process as well as capture and avoid some of the aforementioned artifacts and bias. Finally, we show that the quality and thoroughness of guidelines and of the corpus construction itself play an important role in the amount of disagreement.

2. The Experiment

Our experiment was undertaken with the help of nine Computer Science and Linguistics graduate students in a Computational Linguistics seminar. These annotators were not under time pressure, they did not have a financial motive and had a much smaller number of pairs to work with than an average crowdworker. The students were split in three teams of three, each receiving the same set of 100 NLI pairs but with different annotation guidelines. The goal was to observe whether the different guidelines lead to different amounts of disagreement or whether some pairs have consistently lower IAA scores across guidelines. The students were asked to provide a label for each pair, but also to justify their decision in a short comment. Additionally, they had to give a *Difficulty Score* from 1 to 5 for how hard the annotation of each pair was for them; 1 being very easy and 5 very hard. Thus, the *Difficulty Score* is a subjective human rating score, similar to the common semantic similarity score given to inference pairs (cf. [Marelli et al. 2014]).

The dataset The dataset used for this experiment was constructed with pairs originating from the SICK [Marelli et al. 2014] and the SNLI [Bowman et al. 2015] corpora. SICK is an English corpus of almost 10,000 pairs, annotated for their degree of similarity and for the inference relation between the sentences of each pair. SNLI is an English corpus of over 550,000 pairs, annotated for inference. Both corpora were created from captions of pictures, talking about daily activities and non-abstract entities. SICK was also further simplified in terms of linguistic phenomena included, e.g., named entities and temporal phenomena were removed. From each corpus, we selected 50 pairs: 20 were originally annotated as contradictions (C), 20 as neutrals (N) and 10 as entailments (E). We chose this distribution of labels because controversies are most common in pairs labelled as contradictions and neutrals [de Marneffe et al. 2008, McCoy et al. 2019, Kalouli et al. 2019] and thus it makes sense to include more of these examples to test our hypothesis. From these 50 pairs, 25 were pairs we considered *clear-cut*, i.e., easy, unambiguous inferences where we expect people to agree on (10 Cs, 10 Ns and 5 Es), and 25 were *controversial*, i.e., some ambiguity within the pair could potentially lead to different annotations (10 Cs, 10 Ns and 5 Es). To capture the notion of controversy, we use the findings of [Kalouli et al. 2019]: pairs with *directionality, coreference* and *loose definitions* phenomena are considered controversial. Examples of each type are given in Table 1. At this point, we should clarify that this distinction is *annotation-based*. This means that the *clear-cut vs. controversial* distinction concerns the difficulty of the annotation of the pair and not the linguistic complexity of the pair. This is important because recent work [Glockner et al. 2018, Nie et al. 2018, Dasgupta et al. 2018, McCoy et al. 2019, in-

	Clear-cut	**Controversial**
E	P: A man is riding a horse on the beach. H: A guy is riding a horse.	P: A woman is making a clay pot. H: An artist is sculpting with clay. *is everyone who is making something with clay an artist?*
C	P: A woman holding a boombox. H: A man holding a boombox.	P: A man is holding a small animal in one hand. H: A man is holding a big animal in one hand. *depending on the judge, an animal might be small or big*
N	P: A woman is running a marathon in a park. H: The woman is running fast.	P: A woman is being kissed by a man. H: A lady is being kissed by a man. *is lady a synonym to woman or not?*

Table 1. Examples of *clear-cut* and *controversial* pairs.

ter alia] has shown how there can also be *easy* and *hard* inferences, based on the complexity of the linguistic phenomena involved in the sentences. For example, sentences with modals, passives, word-order scrambling, implicative and factive verbs, etc., are considered hard because models struggle with them. This distinction between *linguistic phenomena* and *annotation* difficulty is essential in our experiment.

The Guidelines As mentioned above, each group of annotators was given different guidelines to deal with the task.[1] The goal behind this strategy is to see if different guidelines lead to more or less controversy and whether there are pairs that are inherently more ambiguous across guidelines. Group 1 received the original SNLI guidelines. These guidelines provide a caption of a picture as the premise and ask the crowd workers to write a sentence that is a definitely-true/definitely-false/might-be-true description of that picture/caption. Since now we already have the P-H pairs, we reformulated these guidelines: the annotators were asked to judge whether H was a definitely-true/definitely-false/might-be-true description of P, as the SNLI creators also did in their validation stage. Group 2 received the improved guidelines proposed by [Kalouli et al. 2019][2] (KAL guidelines). These guidelines attempt to address issues found in the original SICK and SNLI guidelines, e.g., tackling coreference phenomena. The annotators are asked to imagine P as a caption of a picture, describing whatever is on that picture; P represents the truth based on which they have to judge H. Finally, Group 3 was not given any guidelines.[3]

3. Results and Discussion

The set-up of our experiment allows for different kinds of observations. Here, we mainly focus on the most prominent results. The overall goal of the experiment was to test

[1]The exact guidelines will become available after publication.

[2]Available under `https://github.com/kkalouli/SICK-processing`.

[3]They were only given the following: *You get two pieces of text: a premise and a hypothesis. For some examples, the hypothesis follows from the premise ("entailment"). In other cases, the text and the hypothesis are contradictory ("contradiction") and in some others the hypothesis neither follows from nor contradicts the text ("neutral"). Annotate each pair with an inference label, i.e., E for entailment, C for contradiction or N for neutral.*

Group: Guidelines	Min	Max	Mean	St.Dev.	# Clear-cut	# Controversial
Group 1: SNLI guidelines	1	4	2.14	0.71	80	20
Group 2: KAL guidelines	1	4	1.95	0.75	86	14
Group 3: No guidelines	1	2.6	1.5	0.48	71	29

Table 2. The min, max, mean and standard deviation of the *Difficulty score* for the pairs, as labeled by the annotators, and the number of *clear-cut* vs. *controversial* pairs based on the z-score computation.

	Own Classification		Annotators' Classification	
Group: Guidelines	Clear-cut	Controversial	Clear-cut	Controversial
Group 1: SNLI guidelines	72.1	51.9	72.1	20.6
Group 2: KAL guidelines	76.7	52.5	70.5	35.3
Group 3: No guidelines	50.8	38.9	51.5	23.3

Table 3. IAA between *clear-cut* and *controversial* pairs, across groups and guidelines, based on both classification schemes.

whether the controversial pairs correlate with low IAA and high *Difficulty Scores*. In other words, we wanted to test whether the IAA is statistically worse in controversial pairs. At the same time, we wanted to test what effect different guidelines have on this aspect. To this end, we split the pairs into *clear-cut* vs. *controversial* in two different ways: on the one hand, we relied on our own initial classification of the pairs into *clear-cut* vs. *controversial* (cf. Section 2). On the other hand, we used the annotators' *Difficulty Score* to get a notion of ambiguity and controversy: for each group, annotator and pair, we applied z-score normalization of the *Difficulty Score* to account for different raters using the scale differently; then, if the z-score of a given pair was greater than 1, the pair was considered *controversial* and, if it was equal or less than 1, it was considered *clear-cut*. The minimum, maximum, mean and standard deviation of *Difficulty Score* for each group, i.e. for each set of guidelines, is shown in Table 2. We also show the number of pairs classified as *clear-cut* or *controversial* with this process (the exact pairs with their classifications will be available after publication). Based on these two classifications of the pairs, we calculated the IAA between *clear-cut* and *controversial* pairs, across the three sets of guidelines. Results are shown in Table 3.

Despite the small-scale of this experiment, the results lead to enlightening observations. First, we observe that the clear-cut pairs have a significantly higher IAA ($p < 0.05$) than the controversial pairs, both in ours and in the annotators' classification. Interestingly, the significance level is higher for the annotators' classification, i.e., the agreement for the controversial pairs as defined by the annotators themselves is much worse than the agreement for the controversial pairs as defined by our classification. This finding is in line with [Pavlick and Kwiatkowski 2019], who show that disagreements are not to be dismissed as annotation noise, but rather persist as more ratings are collected and as the amount of context provided to raters increases. Similarly, our experiment shows that some disagreements are persistent, no matter the guidelines and the task definition; those are inherent disagreements in the way humans perceive semantic notions and deal with world knowledge. They can also not be solved by using a graded scale of annotation rather than distinct labels: the extent of disagreement will always persist. However,

this does not mean that the task is unsolvable, but rather that a different perspective is required.

One solution is proposed by [Pavlick and Callison-Burch 2016], who show that current SOTA models do not capture the same distribution over inference labels as that of the human judgments. Thus, they argue that NLI evaluation should explicitly incentivize models to predict distributions over human judgments. This solution is useful but does not tackle the issues presented above: the artifacts of the datasets and the generalization difficulties of the models. Concerning the artifacts, current datasets have been shown to have strong correlations between labels and words [Gururangan et al. 2018, Poliak et al. 2018], e.g. contradictions have a strong correlation with the words *no, not, nobody, sleep*, etc., so that models only pick up such statistical patterns rather than the reasoning rules behind the sentences. As far as the generalization difficulty is concerned, current models struggle with complex linguistic phenomena such as compositionality, negation, modals, passives, factives and implicatives, quantifiers, etc. [Nie et al. 2018, Dasgupta et al. 2018, McCoy et al. 2019, Richardson et al. 2019]. However, such *linguistically-hard* phenomena can be *annotation-clear-cut* and unambiguous for humans. For example, a pair like *P: The man chased the cat. H: The cat was chased by the man* or *P: The man walked the dog. H: The dog walked the man* is very easy for humans to annotate but contains compositionality rules which make it difficult for models to get right. Thus, our proposal attempts to also address these challenges: the NLI annotation should be complemented by a *Difficulty Score* like the one introduced here. This score can then serve two roles. First, it can be exploited during the training process: pairs that are *clear-cut* are more reliable for training and should thus have a stronger learning effect, e.g., have higher training weights, than *controversial* pairs with lower IAA. The score can also be exploited during the evaluation process by measuring performance on *clear-cut* vs. *controversial* pairs: it is expected that current SOTA models will fail on many of what might be *annotation-clear-cut* pairs (for humans) but *linguistically-hard* pairs (for machines), and thus this will reflect better the real reasoning power of these models. Second, our proposal can help reduce the artifacts of the datasets. For example, if pairs containing the word *sleep* in H are always judged as contradictory and *clear-cut*, no matter the complexity of P (due to the artifact that sleeping is used to contradict any other action), they can be recognized as artifacts and thus be removed or dealt with otherwise. Of course, consistent human explanations would be even better than the proposed score. However, these tend to be very expensive (if at all possible) during and after the corpus construction [Camburu et al. 2018]. Thus, a score seems a suitable way to mitigate the issues with annotation unevenness, if one is embarking on an annotation project. The alternative ways of generating explanations e.g. LIME (Local Interpretable Model-agnostic Explanations [Ribeiro et al. 2016]), NILE (Natural-language Inference over Label-specific Explanations [Kumar and Talukdar 2020]) or REXC (Rationale-inspired Explanations with Commonsense [Majumder et al. 2021]) from already annotated corpora, seem less direct and less trustworthy. Thus, the proposed difficulty score is one of the main contributions of this paper that we expect to impact how future annotation efforts and NLI datasets are conceived and executed.

Additionally, despite the small scale of this experiment and the inconclusive picture we get, we do observe that guidelines play an important role: Group 1, which was given the SNLI guidelines, has the lowest IAA for the *controversial* pairs in the classifi-

cation done by the annotators' themselves (20.6 vs. 35.4 and 23.3). On the other hand, Group 2 which is given the KAL guidelines, has the fewest *controversial* pairs (14 vs. 20 and 29) and the best IAA for these controversial pairs in both classification schemes (in the annotators' classification the difference is even statistically significant). Group 3 has the most *controversial* pairs and the lowest IAAs both for *clear-cut* and *controversial* pairs in both classification schemes – except for the *controversial* in the annotators' classification. From these findings, we can conclude that a) the nature of the SNLI guidelines leaves much room for interpretation and does not avoid some of the controversy which seems avoidable given the KAL guidelines, b) the SNLI guidelines do not address harder cases and thus there is high disagreement for the annotation of the controversial pairs, c) no guidelines whatsoever (Group 3) do lead people to think that the annotations are easy, presumably because they are not given "restrictions" based on which they should judge the pair (lowest mean Difficulty Score), but no guidelines clearly lead to poor IAA and thus the claim that people should be annotating as naturally as possible [Manning 2006] cannot be justified.

4. Conclusions

In this work we presented an NLI annotation experiment, aimed at investigating whether specific NLI pairs are inherently more difficult to annotate and thus lead to lower IAA. The experiment considered this question given different guidelines, to test to what extent the annotation difficulty is due to the guidelines quality. The results of this work show the value of augmenting the NLI annotation task with a *Difficulty Score* and the ways in which this score can be beneficial. Future work will seek to scale up these findings with a larger-scale experiment and confirm further preliminary findings of the current experiment. We also aim to reproduce the experiment with languages different from English, with corpora as [Fonseca et al. 2016] and [Real et al. 2018] for Portuguese.

References

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Camburu, O.-M., Rocktäschel, T., Lukasiewicz, T., and Blunsom, P. (2018). e-SNLI: Natural language inference with natural language explanations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc.

Dasgupta, I., Guo, D., Stuhlmüller, A., Gershman, S. J., and Goodman, N. D. (2018). Evaluating compositionality in sentence embeddings. *CoRR*, abs/1802.04302.

de Marneffe, M.-C., Rafferty, A. N., and Manning, C. D. (2008). Finding contradictions in text. In *Proceedings of ACL-08*.

de Marneffe, M.-C., Simons, M., and Tonhauser, J. (2018). Factivity in doubt: Clause-embedding predicates in naturally occurring discourse (poster). *Sinn und Bedeutung 23*.

Fonseca, E. R., dos Santos, L. B., Criscuolo, M., and Aluísio, S. M. (2016). Assin: Avaliação de similaridade semântica e inferência textual. In *Proceedings of PROPOR*, pages 1–8.

Glockner, M., Shwartz, V., and Goldberg, Y. (2018). Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655. Association for Computational Linguistics.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Kalouli, A.-L., Buis, A., Real, L., Palmer, M., and de Paiva, V. (2019). Explaining simple natural language inference. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 132–143, Florence, Italy. Association for Computational Linguistics.

Kalouli, A.-L., Real, L., and de Paiva, V. (2017). Correcting Contradictions. In *Proceedings of Computing Natural Language Inference (CONLI) Workshop, 19 September 2017*.

Kumar, S. and Talukdar, P. (2020). NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online. Association for Computational Linguistics.

Liu, X., He, P., Chen, W., and Gao, J. (2019). Multi-task deep neural networks for natural language understanding. *CoRR*, abs/1901.11504.

Majumder, B. P., Camburu, O., Lukasiewicz, T., and McAuley, J. J. (2021). Rationale-inspired natural language explanations with commonsense. *CoRR*, abs/2106.13876.

Manning, C. D. (2006). Local textual inference: it's hard to circumscribe, but you know it when you see it–and nlp needs it. *https://nlp.stanford.edu/manning/papers/Textual Inference.pdf*.

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC 2014*.

McCoy, T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Nie, Y., Wang, Y., and Bansal, M. (2018). Analyzing compositionality-sensitivity of NLI models. *CoRR*, abs/1811.07033.

Palomaki, J., Rhinehart, O., and Tseng, M. (2018). A case for a range of acceptable annotations. In *SAD/CrowdBias@ HCOMP*, pages 19–31.

Pavlick, E. and Callison-Burch, C. (2016). Most "babies" are "little" and most "problems" are "huge": Compositional entailment in adjective-nouns. In *Proceedings of*

the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2164–2173, Berlin, Germany. Association for Computational Linguistics.

Pavlick, E. and Kwiatkowski, T. (2019). Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Real, L., Rodrigues, A., e Silva, A. V., Albiero, B., Guide, B., Thalenberg, B., Silva, C., Câmara, I. C. S., de Oliveira Lima, G., Souza, R., Stanojevic, M., and de Paiva, V. (2018). Sick-br: a portuguese corpus for inference. In *PROPOR 2018*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Richardson, K., Hu, H., Moss, L. S., and Sabharwal, A. (2019). Probing natural language inference models through semantic fragments. *CoRR*, abs/1909.07521.

Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., and Zhou, X. (2019). Semantics-aware BERT for language understanding.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

A machine learning approach to literary genre classification on Portuguese texts: circumventing NLP's standard varieties

Dionéia Motta Monte-Serrat, Mateus Tarcinalli Machado
Evandro Eduardo Seron Ruiz

[1]Department of Computing and Mathematics – FFCLRP
University of São Paulo – Ribeirão Preto, Brazil

di_motta61@yahoo.com.br, {mateusmachado, evandro}@usp.br

Abstract. *We evaluate and classify bottom-up and quali-quantitatively literary genres from the BDCamões corpus. Chronicles, novels, short stories, and tales, annotated in UD, are classified by random forests and analyzed based on the Brazilian-Portuguese version of the LIWC dictionary. Results by class are reported by the mean value, along with a measure of variability. The results for features by class, LIWC tags, part of speech, and Universal Dependency tags highlight the higher positive and lower negative features. Adapting this method to the fluidity and mutability of literary genres circumvents the difficulty of NLP's standard tools, showing consistency and fewer errors in the results.*

Resumo. *Avaliamos e classificamos quali-quantitativamente gêneros literários do corpus BDCamões. Crônicas, romances, histórias curtas e contos, anotados em UD, são classificados por florestas aleatórias, e analisados com base na versão português-brasileira do LIWC. Os resultados por classe são reportados pela média, juntamente com uma medida de desvio padrão. Os resultados das características por classe, rótulos LIWC, classes gramaticais e rótulos UD destacam características positivas altas e negativas baixas. A adaptação desta metodologia à fluidez e mutabilidade dos gêneros literários contorna as dificuldade normalemnet encontradas em NLP, apresentando consistência e poucos erros nos resultados.*

1. Introduction

Classifying documents according to their genre is proposed as a primary task in text and document processing. Eissen and Stein [Sven Meyer zu and Stein 2004] claim genre classification is usually performed by discriminating documents through their form, their style, or their targeted audience. They also assure that genre classification is orthogonal to a classification based on the documents' contents. This study considers the artistic content of the analyzed texts since it focuses on literary genres, which materialize the style or form adopted by the authors, being, therefore, closely linked to the organizational goals and processes through which writing went.

The textual genre has a broad conception due to the socio-communicative function of documents, which encompasses the intention, context, social function, among other aspects of the document [Martins 2008]. Text is formed from expressed ideas. These ideas, structured in textual types, are also determined from interconnected factors, such as socio-historical context, physical and subjective structures of individuals,

grammatical rules, social function, and interlocutors. This multiplicity of factors is materialized, in an unstructured way, in messages such as posts on social networks, scientific articles, books, news, reports, and others, giving rise to different textual genres. This article is dedicated to studying the classification of literary genres, a specific text classification task within natural language processing (NLP). Textual genres usually rely on their differences in textual structure and leave behind a wide range of contextual factors. The challenge in analyzing literary genres lies in non-structured data from context inputs that blend into words and expressions to carry meaning to the grammatical reality of the text [Bronckart 2004, Monte-Serrat 2021, Monte-Serrat and Cattani 2021b, Monte-Serrat and Cattani 2021a]. Textual genres are fluid and changeable, continuously adapting to new social needs [Bronckart 2004, Matos 2021]. Suppose the focus falls on only one of the features (only on a contextual feature or only on the human language's logic/grammatical feature). In that case, the differentiation process might be deficient, carrying weakness that may be transmitted to the final findings.

Deepening the knowledge about literary genres classification can improve web search and information access to extensive digital collections [Crowston and Kwasnik 2003, Karlgren et al. 1998]. If not only to improve information retrieval, distinguishing different literary genres is a demanding task. The difference between poetry and a novel may be evident for an educated human being, but it may not be easy to discriminate a short story from a chronicle.

The main objective of this work is to study quantitative features that may differentiate literary genres under the scope of the NLP subject. We aim to address some corpus annotation features that may help find the regularity underlying the literary genres of the corpus of the BDCamões [Grilo et al. 2020]. This article is divided into five sections. Section 2 deals with related research on textual genre identification, showing that genre knowledge makes information more easily understood by search tools. Section 3 describes data and methods, discussing the corpus of textual genres BDCamões annotated according to the Universal Dependency framework. The results are described in Section 4, where they are segmented by classes reported by the mean value and its standard deviation. We also highlight the importance of features per class, LIWC labels, part-of-speech and Universal Dependency labels. Section 5 addresses the discussion of the results, showing from a bottom-up perspective how these NLP tools adapt to the contextual characteristics of fluidity and mutability of literary genres. We conclude in Section 6 that the analysis of literary genres must contemplate their artistic content to identify the aesthetics of the 'best order or structure' of words or the aesthetics of the 'best word'. For this, we suggest a combination of tools that balance structural and contextual aspects, making the machine more intuitive with more consistent results.

2. Related work

Some papers focus on genre identification for information retrieval in extensive digital collections. When characterized only by the textual form, this identification is not enough to define an information problem. This identification is due to the interdependent relations, in textual genres, between linguistic units and contextual parameters [Schneuwly 1997]. Search results containing genre information are easier understood by search tools, as the genre is often an implicit notion.

Karlgren and collaborators [Karlgren et al. 1998] make iterative information retrieval through topic grouping to build a multidimensional pre-representation interface of the research results. This way, the authors enrich the information search dialogue, encouraging and supporting the iterative refinement of queries, and enrich the document representation beyond the simple semantics of the terms frequencies.

Statamatos and co-authors [Stamatatos et al. 2000] use word frequency from The Wall Street Journal training corpus. Compared to the most frequent words in plain English cited in the British National Corpus, the authors claim that the latter contains more reliable discriminators for classifying text genres than the most frequent words in their limited-size training corpus. Similarly, Feldman and his team [Feldman et al. 2009] proposed using part-of-speech (POS) histogram statistics to perform the classification of textual genres. Together with a quadratic discriminant classifier, they show better performance than techniques that use word frequency counting features and POS tri-gram features. The authors claim that it is unclear what techniques would be needed to cover the entire feature space and differentiate the sub-classes, suggesting to characterize the genre more generalizable, as a multitasking learning, replacing singular genre classes with multiple factors.

Nilan et al. [Nilan et al. 2001] employ a bottom-up approach to analyze perceptions of textual resources that assist users in characterizing documents. Using content analytic techniques, the authors derive a set of genres built around the actual use of the web to compare existing genre lists. Mark Rosso [Rosso 2005] reports a study in which users classify genres according to a palette for use in web retrieval. Each participant received a pile of 102 printed web pages and was asked to separate the pages into piles according to the genre. The level of agreement reached 60%. In another study, the author analyzes 18 genres in an online experiment in which 257 subjects. The agreement rate reached more than 70% regarding the textual genre.

Omar [Omar 2020] brings together the Vector Space Clustering (VSC), ‘concept pool’ (BOC), explicit semantic analysis (ASE), and ConceptNet methods to address the classification of literary genres. They show that the computational and semantic models approach results to achieve better performance in the classification task. The author claims that the dimensionality of the data makes it difficult to obtain reliable analytical results, suggesting classification only in the most critical or distinct resources available.

3. Data and Methods

Data

The corpus of textual genres used in this research is the BDCamões Collection of Portuguese literary documents [Grilo et al. 2020]. Some features of the BDCamões corpus make it very useful for research in NLP: it is composed of 4 million words from more than 200 complete documents written by 83 authors in 14 genres. Its literary texts range from the 16th to the 21st century and have been carefully edited. The dimensionality of the data makes it difficult to obtain reliable analytical results [Omar 2020]. Monte-Serrat and Cattani [Monte-Serrat and Cattani 2021a] mention the curse of dimensionality in data interpretation. These arguments support the choice of the BDCamões corpus to find reliable results.

The corpus includes automatically annotated linguistic information such as grammatical classes, morphological features, grammatical dependencies based on the Universal Dependency framework (UD) [Nivre 2015], and expressions denoting named entities. BDCamões brings classified texts according to the following literary genres: 92 tales; 26 chronicles; 25 novels; 21 short stories; 18 poems; 11 theater plays; 8 essays; 1 travel guide; 1 sermon; 1 other; 1 narrative; 1 memoir; 1 letter; 1 anthology, totaling 208 documents and 3,945,943 words.

Methods

We chose random decision forests [Breiman 2001], an ensemble learning method, for the literary genre classification task. Random forest is a popular machine learning algorithm consisting of a combination of tree classifiers. Each classifier is generated using a random vector sampled independently from the input vector. Every tree casts a single vote, choosing the most popular class to classify an input vector. Decision trees seek to find the best split to subset the data based on the features provided to the learning phase.

Linguistic Inquiry and Word Count (LIWC) [Tausczik and Pennebaker 2010] is a text analysis system created by Pennebaker and collaborators [Pennebaker et al. 2001] with the aim of grouping words into categories that can be used to analyze psycholinguistic characteristics in different types of texts, making this tool interesting for the assessment of literary genres. LIWC is composed of software tools and a lexicon/dictionary. Each LIWC dictionary entry can be assigned to one or more categories (The word 'like' can belong to the category 'pronoun' or 'discrepancy' or 'affection' or even 'simile'). LIWC includes 17 standard linguistic dimensions (e.g., word count, percentage of pronouns, articles), 25 word categories tapping psychological constructs (e.g., affect, cognition), 10 dimensions related to "relativity" (time, space, motion), and 19 personal concern categories (e.g., work, home, leisure activities)': "LIWC successfully measures positive and negative emotions, a number of cognitive strategies, several types of thematic content, and various language composition elements" [Pennebaker et al. 2015]. The core of this program is known as the LIWC dictionary, which was made available for the Brazilian-Portuguese. In this research, we used this Brazilian version of LIWC 2007 [Balage Filho et al. 2013] to present the best and worst combinations of categories in analyzing texts from the BDCamões corpus.

BDCamões delivers different numbers of classified texts according to literary genres. To pursue consistent results, we selected only the literary genres that had more than ten text samples. These were: tale (96), novel (25), short story (21), chronicle (26), and poetry (18). As mentioned before, the corpus was already annotated with part-of-speech and UD labels. We used these annotations as features for the random forest classifier, and we also added word categories labels obtained from the Brazilian Portuguese LIWC [Balage Filho et al. 2013]. As for the latter, this dictionary/lexicon is composed of 64 word classes. Many words have multiple class labels. In this case, all word labels were added as features for the classifier. We did not use words as a feature, as our goal is to seek a categorization of texts focused on their structure, not on the content.

A grid search method was applied to find the best parameters to train a random forests model. For this, we used stratified 3-fold cross-validation together with a total of 4,320 combinations of parameters. The best combination of parameters was chosen by

calculating the F-measure. Once the best parameters were found, we configured a new classifier model using these parameters with a stratified 5-fold cross-validation scheme.

In order to analyze and understand the importance of the selected characteristics in the classification, we used the Python language module Eli5[1]. Eli5 allows the explanation of weights and predictions made by machine learning models. The weights of each attribute are calculated by following decision paths in all trees created by the classification model. Each node of the tree has an output score. The feature contribution on the decision path is calculated using the score difference from a parent to child node. Weights of all features sum to the output score or probability of the estimator.

4. Results for literary genre classification

The results based on a classification of resources' average among all classes of literary genres make the data more similar to the expected target domain of each genre. See Table 1. This table implies a balance between two aspects that literary genres commonly bring embedded: i) the assessment that considers the textual type (in which what matters most is the structural organization providing specific sequences for narration, description, exposition, argumentation) [Marcuschi et al. 2002]; ii) and the assessment of the socio-historical aspects, text function, media, type, and adequacy of language, among others, which influence that textual type.

Table 1 shows the results for literary genre classification based on the comprehensive set of part-of-speech, UD, and LIWC features. The poetry genre obtained the best classification scores for precision and recall measures among the five genres tested, reaching a harmonic mean of 88% (SD 11%). Although the chronicle genre obtained 100% precision, 29% (SD 29%) recall contributed to the second-lowest F-measure (24%) among all the genres. Novels also obtained inferior results, which reflected an F-measure of 11%. Contrary to our initial guess, tales and short stories might have very few in common regarding these classification features. Tales presented an F-measure of 77% with the lowest SD of 7%. On the opposite side, short stories presented higher percentages of standard deviation for precision and recall, matching the final F-measure to its SD.

The weighted average is a metric computed this way: find the corresponding metric's average for each class weighted by the number of true instances for each label. Then compute the average among all these classes. The weighted average values for precision, recall, and F-measure reflect the inconsistency and the variety of the classes' metrics.

The common ground on what constitutes a domain is something idealized. We recall that for Plank [Plank 2011] the common ground does not exist. Plank affirms that the literary genre can be considered a domain that mixes those textual types and socio-historical aspects. While Table 1 idealizes the pattern of these aspects for each genre, we can observe that poetry and tale present outstanding results ($\overline{x}$ =0.88 and 0.66; F-measure=0.88 and 0.77, respectively). We infer this highlight occurs because their textual type stands out to their socio-historical aspects. This result is not repeated with chronicles, novels, and short story. The socio-historical aspects of the latter are presented in greater proportion compared to the textual type, which makes the normalization process more difficult to reduce their differences (F-measures=0.24; 0.11; 037, respectively, as

[1] https://eli5.readthedocs.io/en/latest/

shown in Table 1). When we address Table 2 we will clear these assertions. The analysis of chronicles, novels, and short stories becomes challenging because of the uncertainties in the choice or extension of the characteristics of each of these genres, which turns into a contradiction if studies by two or more literary critics were compared. Literary genre is an ongoing problem that the constitutive classification weaknesses of the gender notion [Altman 1984]. The tool reflects this weakness when displaying the F-measure for chronicle, novel, and short story, respectively.

Table 1. Results per class reported by the mean (average value) along with a measure of variablility (SD, standard deviation).

Class	Precision		Recall		F-measure	
	Mean	SD	Mean	SD	Mean	SD
Chronicle	**1.00**	0.00	0.18	0.29	0.24	0.36
Novel	0.40	0.42	0.08	0.11	0.11	0.16
Poetry	**0.87**	0.19	**0.93**	0.15	**0.88**	0.11
Short story	0.73	0.43	0.30	0.33	0.37	0.37
Tale	0.66	0.08	**0.92**	0.06	**0.77**	0.07
Weighted average	0.57	0.12	0.66	0.07	0.58	0.09

Table 2 focuses on the feature importance per class, permitting to investigate the artistic content of each literary genre [Marcuschi et al. 2002]as they are a set of works of the same nature, with essentially identical trends [Almeida 2005], linked to similar cultural periods. Features linked to the notion of time stand out in poetry, novels, and chronicles, such as: `LW:time` 0.036; 0.025 for poetry and novel respectively; and `LW:past` 0.019, 0.016 and 0.015 for chronicle, poetry, and novel respectively. These features appear as the lowest negative features for short stories and tales: `LW:past` -0.006, -0.044 and `LW:time` -0.053 respectively, confirming that these two genres focus more on structure than narrative, which is evidenced in the prevalence of morphosyntactic annotation of universal dependence and POS aimed at analyzing the linguistic features of a word along with its preceding as well as following words (PO and UD tags)

5. Discussion

How the authors of the texts under analysis write provide clues to emotion and cognition [Gottschalk and Gleser 1979, Rosenberg and Tucker 1979]. The LIWC dictionary [Balage Filho et al. 2013] offers an efficient method to study the emotional components of literary works, going beyond the analyzes that commonly focus on the structure of those texts. We demonstrate that the analysis becomes more precise using a context-motivated tool (taking the classification of textual genres as contextual information), bringing results closer to state-of-the-art. Our strategies offer results with potential uses to set limits on the accuracy, being easy to replicate and interpret: i) the use of LIWC provides contextual data that make the tool more intuitive; the inclusion of all LIWC categories reduces complexity and facilitates tool optimization; ii) each node of the Random Forest reduces the built-in freedom of context, softening the dispersion of available information; iii) Part-of-Speech label words identifying their function (grammatical class tag such as noun, verb, article, adjective, preposition, pronoun, adverb, conjunction and

Table 2. Feature importance per class. LW, PO and UD stand for LIWC, part-of-speech and Universal Dependency labels, respectively.

Chronicle		Novel		Poetry		Short story		Tale	
Feature	Weight	Feature	Weight	Feature	Weight	Feature	Weight	Feature	Weight
Ten highest positive features									
LW:past	0.019	LW:time	0.025	LW:time	0.036	LW:ingest	0.006	PO:UM	0.013
LW:assent	0.009	UD:NSUBJ	0.017	LW:past	0.016	PO:LADV1	0.005	UD:POBJ	0.012
LW:quant	0.008	LW:past	0.015	LW:preps	0.009	LW:percept	0.005	UD:CASE	0.010
PO:LTR	0.008	LW:home	0.015	PO:LTR	0.006	LW:see	0.004	UD:NUMBER	0.009
UD:CC	0.003	PO:LTR	0.015	UD:CASE	0.005	UD:CC	0.002	PO:PNT	0.009
LW:motion	0.002	LW:see	0.013	LW:sad	0.005	PO:LADV2	0.002	UD:ROOT	0.009
LW:ingest	0.002	LW:assent	0.012	LW:motion	0.004	PO:LPREP1	0.002	LW:cogmech	0.007
PO:GER	0.002	PO:LITJ2	0.012	UD:POBJ	0.004	LW:i	0.002	UD:MWE	0.007
LW:conj	0.002	UD:PREDET	0.010	LW:assent	0.003	LW:health	0.001	LW:bio	0.006
LW:see	0.002	LW:sad	0.010	LW:ingest	0.002	PO:DGT	0.001	UD:PUNCT	0.005
Ten lowest negative features									
UD:CASE	-0.006	PO:UM	-0.003	LW:cogmech	-0.012	UD:POBJ	-0.009	LW:time	-0.053
UD:POBJ	-0.005	UD:ROOT	-0.003	LW:funct	-0.009	UD:CASE	-0.008	LW:past	-0.044
LW:preps	-0.005	LW:future	-0.001	LW:cause	-0.008	PO:EOE	-0.008	PO:LTR	-0.029
LW:time	-0.005	UD:POBJ	-0.001	LW:tentat	-0.007	UD:IOBJ	-0.006	LW:assent	-0.024
PO:DA	-0.003	UD:CASE	-0.001	LW:inhib	-0.006	LW:past	-0.006	LW:see	-0.019
PO:PNT	-0.003	UD:NUMBER	-0.001	LW:adverb	-0.006	UD:NUMBER	-0.006	LW:ingest	-0.018
UD:NSUBJ	-0.003	LW:bio	-0.001	LW:quant	-0.006	PO:UM	-0.005	UD:CC	-0.015
PO:CARD	-0.003	PO:LADV4	0.000	PO:UM	-0.005	PO:PNT	-0.005	LW:home	-0.014
PO:ORD	-0.003	UD:PUNCT	0.000	PO:VAUX	-0.005	UD:MWE	-0.004	PO:LITJ2	-0.012
UD:MARK	-0.003	PO:LITJ1	0.000	LW:conj	-0.004	LW:bio	-0.004	UD:NSUBJ	-0.012

interjection) from the relationship with relative terms and by definition (probability-based and rule-based) help to reduce ambiguity.

For the approaches to literary genres to achieve results in state-of-the-art, it is necessary to adapt the tool to the fluidity and mutability characteristics of these genres, making the machine obey the rules of the nature of the analyzed text. These are the rules/strategies that we try to expose in this research work. Working bottom-up, the system establishes the best and worst feature combinations to classify specific literary genres.

According to Lüthi [Lüthi 1970] tales follow a distinct style for unfolding the genre in lasting appeal to people. Tale's unique style of structure, symbolism, and meaning offer "sharpness and precision" because it eliminates most descriptions (prevalent in chronicles, novels, poetry, and short stories), giving tales a universal meaning that opens up an opportunity for the use of the imagination. Therefore, we found consistency with the results in Table 2, as the absence of those descriptive details gives greater importance to the text structure than to the psychological characteristics, reducing the efficiency of the LIWC.

The everyday basis of the narrative structure present in the chronicles, short stories, and novels did not 'deceive' the tool, giving less weight to the LIWC attributes for the short story. This result in Table 2 is consistent because the linguistic sequences of the short story are more streamlined than the chronicle and the novel, increasing the importance of features related to the structure of the text relatively to the psycholinguistic attributes of the LIWC. The short story is based on the principle of offering a faster reading than a novel. Therefore, it condenses information, reduces the number of facts presented, and aesthetic strategies meet this need.

It is important to emphasize that the novel genre can contain several textual types, making the tool's evaluation perform very poorly. In some cases, it is possible to find

genres with a specific typology, such as poetry, which improves the performance of the analysis. As seen in Table 1.

Time is a feature that stands out in romance and poetry. This highlight in the novel is due to the intrinsic narrative situated in time. However, the tool also found greater prominence for the feature time in poetry. See Table 2. This finding is justified because poetry evokes an imaginative awareness, organizing its meaning, sound, and rhythm through language [Nemerov 2020]. There is a hypothetical expression of things in poetry that stands out from the storytelling of facts in the novel. In poetry, contemplation evokes feelings, leading the reader to a delight intrinsic to art (the Beauty) so as not to 'freeze' the senses in separate classes of objects. In this way, poetry acts on the human spirit, becoming recognizable because it depends on a line as a parameter, which guides the reading through the displacement of the latter concerning breathing and syntax [Nemerov 2020]. This characteristic changes its appearance, which makes interpretation by the tool more accurate. This reading process (line) is essential to differentiate the tone or rhythm of poetry from the novel.

The precision-of-meaning effect is greater in the novel than in poetry, making syntax-based tools more obvious to use as they deal with the measurable. Refer to UD tags in Table 2. In poetry, meaning is less accessible to observing the 'best order' to operate in the 'best words', which determines the artistic attitude (the Beauty) concerning definitions in general. This poetic structure has to do with pleasure, with delight in the form of arrangement of sounds about thoughts [Nemerov 2020]. Although poetry deals with commonplace matters, its structure does not have the characteristic of the commonplace. Poetry contains forms of production of inferences like the forms of parables, adapting to the metamorphosis of sentences, transcending the topic dealt with, that is, extending time [Monte-Serrat 2017]. It is inferred, therefore, why the feature time is so important in poetry. See Table 2. These are some comments on considering these elements as strategies to establish the relationship of literary genres with the interpretation to be performed by our tool.

6. Conclusion

We conclude that analyzes of literary genres must consider the artistic content of the texts, as aesthetics are a fundamental element for the various genres. The identification of aesthetics from the search for the 'best order or structure' of words or aesthetics from the search for the 'best word' is of paramount importance, since literary genres materialize the style or form adopted by their authors, linking the writing of the latter to the objectives and organizational processes through which the text went through. The combination of tools that we suggest in this research work provides a textual analysis that balances structural and contextual aspects so that the tool works more intuitively, approaching the state-of-the-art. The cases in which the results were not satisfactory (see Table 1) are due to the complexity of the elements that make up the literary genre. NLP deals with canonical varieties that are considered standard [Plank 2016] and the challenge in analyzing literary genres lies in data variations. Our method suggests how to improve data training. We indicate how best to leverage contextual (literary genres) data that is forgotten and needs to be refined to produce more robust models. It is not about making prescriptions for dealing with textual genres. We make assumptions about which tools are best suited for each genre, training the system to make fewer mistakes.

References

Almeida, N. M. d. (2005). *Gramática metódica da Língua Portuguesa*. Saraiva, 45 edition.

Altman, R. (1984). A semantic/syntactic approach to film genre. *Cinema Journal*, pages 6–18.

Balage Filho, P., Pardo, T. A. S., and Aluísio, S. (2013). An evaluation of the Brazilian Portuguese LIWC dictionary for sentiment analysis. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Bronckart, J.-P. (2004). Les genres de textes et leur contribution au développement psychologique. *Langages*, 1(153):98–108.

Crowston, K. and Kwasnik, B. H. (2003). Can document-genre metadata improve information access to large digital collections? *LIBRARY TRENDS*, 52(2):345–361.

Feldman, S., Marin, M. A., Ostendorf, M., and Gupta, M. R. (2009). Part-of-speech histograms for genre classification of text. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4781–4784. IEEE.

Gottschalk, L. A. and Gleser, G. C. (1979). *The measurement of psychological states through the content analysis of verbal behavior*. University of California Press.

Grilo, S., Bolrinha, M., Silva, J., Vaz, R., and Branco, A. (2020). The BDCamões Collection of Portuguese Literary Documents: a Research Resource for Digital Humanities and Language Technology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 849–854.

Karlgren, J., Bretan, I., Dewe, J., Hallberg, A., and Wolkert, N. (1998). Iterative information retrieval using fast clustering and usage-specific genres. In *Eight DELOS workshop on User Interfaces in Digital Libraries*, pages 85–92.

Lüthi, M. (1970). Once Upon a Time: On the Nature of Fairy Tales. *Trans. Lee Chadeayne & Paul Gottwald. New York: Frederick Ungar Publishing Co*.

Marcuschi, L. A. et al. (2002). Gêneros textuais: definição e funcionalidade. *Gêneros textuais e ensino*, 2:19–36.

Martins, N. S. (2008). *Introdução à estilística: a expressividade na língua portuguesa*, volume 71. Edusp.

Matos, T. (2021). Gêneros textuais. `https://www.portugues.com.br/redacao/generostextuais.html`. Online; accessed July 17th of 2021.

Monte-Serrat, D. (2017). Neurolinguistics, Language and Time: investigating the verbal art in its amplitude. *International Journal of Perceptions in Public Health*, 1(3):162–171.

Monte-Serrat, D. (2021). Operating language value structures in the intelligent systems. *Advanced Mathematical Models & Applications*, 6(1):31–44.

Monte-Serrat, D. M. and Cattani, C. (2021a). Interpretability in neural networks towards universal consistency. *International Journal of Cognitive Computing in Engineering*, 2:30–39.

Monte-Serrat, D. M. and Cattani, C. (2021b). *The Natural Language for Artificial Intelligence*. Elsevier.

Nemerov, H. (2020). Poetry. Encyclopedia Britannica. https://www.britannica.com/art/poetry. [Online; accessed 05-August-2020].

Nilan, M. S., Pomerantz, J., and Paling, S. (2001). Genres from the Bottom Up: What Has the Web Brought Us? In *Proceedings of the ASIST Annual Meeting*, volume 38, pages 330–39. ERIC.

Nivre, J. (2015). Towards a Universal Grammar for Natural Language Processing. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, pages 3–16, Cham. Springer International Publishing.

Omar, A. (2020). Classifying literary genres: a methodological synergy of computational modelling and lexical semantics. *Texto Livre: Linguagem e Tecnologia*, 13(2):83–101.

Pennebaker, J. W., Boyd, R. L., Jordan, K., and Blackburn, K. (2015). The development and psychometric properties of liwc2015. Technical report.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic Inquiry and Word Count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71.

Plank, B. (2011). *Domain Adaptation for Parsing*. PhD thesis, University of Groningen, https://bplank.github.io/publications.html. ISBN: 978-90-367-5199-5.

Plank, B. (2016). What to do about non-standard (or non-canonical) language in NLP. In *Proceedings of the 13th Conference on Natural Language Processing*.

Rosenberg, S. D. and Tucker, G. J. (1979). Verbal behavior and schizophrenia: The semantic dimension. *Archives of General Psychiatry*, 36(12):1331–1337.

Rosso, M. A. (2005). What type of page is this? Genre as Web descriptor. In *Proceedings of the 5th ACM/IEEE–CS joint Conference on Digital libraries*, pages 398–398.

Schneuwly, B. (1997). Textual organizers and text types: Ontogenetic aspects in writing. *Processing interclausal relationships. Studies in the production and comprehension of text*, pages 245–263.

Stamatatos, E., Fakotakis, N., and Kokkinakis, G. (2000). Text genre detection using common word frequencies. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Sven Meyer zu, E. and Stein, B. (2004). Genre classification of web pages. In Biundo, S., Frühwirth, T., and Palm, G., editors, *KI 2004: Advances in Artificial Intelligence*, pages 256–269, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tausczik, Y. R. and Pennebaker, J. W. (2010). The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, 29(1):24–54.

Acknowledgement

This work was carried out at the Center for Artificial Intelligence (C4AI–USP), with support by the São Paulo Research Foundation (FAPESP Grant #2019/07665-4) and by the IBM Corporation.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Evaluation of Synthetic Datasets Generation for Intent Classification Tasks in Portuguese

Robson T. Paula[1], **Décio G. Aguiar Neto**[2], **Davi Romero**[1], **Paulo T. Guerra**[1]

[1]Federal University of Ceará, Campus Quixadá, Ceará, Brazil

[2]Institute of Computation, University of Campinas, São Paulo, Brazil

robson@alu.ufc.br, aguiar@ic.unicamp.br,
{daviromero,paulodetarso}@ufc.br

Abstract. *A chatbot is an artificial intelligence based system aimed at chatting with users, commonly used as a virtual assistant to help people or answer questions. Intent classification is an essential task for chatbots where it aims to identify what the user wants in a certain dialogue. However, for many domains, little data are available to properly train those systems. In this work, we evaluate the performance of two methods to generate synthetic data for chatbots, one based on template questions and another based on neural text generation. We build four datasets that are used training chatbot components in the intent classification task. We intend to simulate the task of migrating a search-based portal to an interactive dialogue-based information service by using artificial datasets for initial model training. Our results show that template-based datasets are slightly superior to those neural-based generated in our application domain, however, neural-generated present good results and they are a viable option when one has limited access to domain experts to hand-code text templates.*

1. Introduction

A chatbot is an artificial intelligence (AI) based system aimed at chatting with users, commonly used as a virtual assistant to help people or answer questions [Al-Sinani and Al-Saidi 2019]. RASA [Bocklisch et al. 2017] is an open-source tool that allows the development of *chatbots* by people who are not experts in this area [Bocklisch et al. 2017]. The tool works by identifying the intentions contained in the user's messages, classifying them according to the intent defined by the developer, and returning a response based on the intent, which may be an already defined text or a custom action, such as access to database [Bocklisch et al. 2017].

Intent classification is essential for a *chatbot* since it must properly identifies what the user wants and, consequently, what the response to be returned by the system. Thus both a good database and a good model are needed. However, a good database for a given task is not always available, especially when the task involves an unexplored domain, making it difficult to train the model.

In this work, we investigated methods to the development of good synthetic datasets with questions about public services of the government of the state of Ceará. The generated data has labels to classify intentions that indicate what type of information the question requires. We proposed methodologies to generate synthetic datasets based on templates and neural generation (using neural networks). To evaluate this generated

data, we trained deep learning models to classify intents to be used these models for the development of *chatbots*.

1.0.1. Related works.

In [Amin-Nejad et al. 2020] the authors propose a methodology to guide the generation with structured patient information in a sequence-to-sequence manner. They propose an experiment with state-of-the-art Transformer models and demonstrate that their augmented dataset is capable of beating baseline models on the downstream classification task.

In [Ive et al. 2020] the authors present an approach to generate artificial medical documents. They propose an approach to discharge summaries from a large mental healthcare provider and discharge summaries from an intensive care unit. They apply several measures of text preservation and how much the model memorizes training data. They estimate the clinical validity of the generated text based on a human evaluation task. They found that using their artificial data as training data can lead to classification results that are comparable to the original results.

Finally, in [Bird et al. 2020] the authors present an approach to the training of deep learning chatbots for task classification called *Chatbot Interaction with Artificial Intelligence* (CI-AI) framework. CI-AI augments human-sourced data via artificial paraphrasing by the T5 model in order to generate a large set of training data for further language learning approaches. The authors show that seven state-of-the-art transformer models are improved when training data is artificially augmented with an average increase of classification accuracy by 4%.

2. Language Understanding and Dialogue Systems

2.1. Language Models based on Transformers

In [Vaswani et al. 2017], the authors proposed the Transformer architecture. This architecture brought a new and simple network architecture based only on attention mechanisms, thus dispensing recurrence and convolutions. They show that machine translation tasks using these models are superior in quality and require significantly less time to train since they can perform a better parallel computation.

2.1.1. BERT

Bidirectional Encoding Representations of Transformers (BERT) is an architecture proposed by [Devlin et al. 2019] used as a pre-trained base model and which can be used to create different state-of-the-art models in natural language processing tasks. This model, based on the use of the transformer encoder proposed by Vaswani [Vaswani et al. 2017]. The BERT model is pre-trained using two tasks, masked language model (MLM) and next sentence prediction (NSP) on an English Wikipedia text corpus and BookCorpus. The model follows a multi-task learning strategy (MTL), where its parameters are trained in a shared way between the two tasks simultaneously, through a shared loss.

BERT is presented in two versions: a base (BERT-base) and a large (BERT-large). BERT-base has 12 attention layers of 12 heads and a hidden layer with 768 neurons totaling 110M of parameters BERT-large has 24 attention layers with 16 heads each and one hidden state of 1024 totaling 340M of parameters.

Given the new state-of-the-art obtained by BERT, was proposed BERTimbau [Souza et al. 2020], a Portuguese version of BERT. This new model based on BERT was pre-trained on a large Brazilian Portuguese corpus named BrWaC (Brazilian Web as Corpus) using the same pre-training method as BERT.

2.1.2. T5

Text-to-Text Transfer Transformer (T5) is a training framework that aims to unify different natural language processing tasks, making the resulting model robust enough to handle a large set of tasks, defining them in the form of sequence-to-sequence problems. This model was developed using an incremental methodology where each step a NLP characteristic was evaluated, such as variable size architecture with respect to encoder and decoder layers and the attention mask that should be used in the model.

The T5 model were trained with unsupervised objectives providing mechanisms by which the model acquires general-purpose knowledge to apply to further tasks. And, similar to BERT, there is a T5 version pre-trained for Brazilian Portuguese data named PTT5 [Carmo et al. 2020].

2.2. Dialogue Systems and Chatbots

Dialogue Systems is a term used to denote systems that provide a conversational interface that allows users to interact with a computer using natural language. Chatbots describe dialogue systems that use text to interact with a user.

Natural language understanding is a key task for dialogue systems. Its main goal is to detect semantic information expressed in the current user utterance. Natural language understanding can be decomposed into different subtasks [Deriu et al. 2021]: (i) identification of domain (if multiple domains), (ii) identification of intents (that is, the question type, the dialogue act, etc.), and (iii) identification of the slots or concept detection. We focus in this work on identification of intents.

2.2.1. Intent Classification.

The NLU task of classifying an utterance into one of the pre-defined intents is called *intent classification* [Chen et al. 2017]. In an utterance such as, "`I want to change my last reservation.`", the intent classifier should identify the utterance intent as `update_reservation` rather than `cancel_reservation` or `new_reservation` intent. The intent classifier could also associate a confidence value to its output, as 0.9 to `update_reservation` and 0.1 to `cancel_reservation`.

Deep learning techniques have been successively applied in intent classification [Dauphin et al. 2013, Deng et al. 2012, Hashemi et al. 2016, Huang et al. 2013, Shen et al. 2014, Tur et al. 2012]. In particular, the RASA team introduces in

[Bunk et al. 2020] a flexible architecture for intent and entity modeling called Dual Intent and Entity Transformer (DIET). DIET outperforms state-of-the-art models even in a purely supervised setup without any pre-trained embedding.

2.3. Synthetic Data Generation Methods

Existing methods for synthetic text data generation can be summarized into two major categories [Peng et al. 2020]: (i) template-based methods that require domain experts to handcraft templates for each domain and a system fills in slot-values afterward; and (ii) statistical language models that learn to generate fluent responses via training from a labeled corpus.

2.3.1. Template-based Generation

As pointed in [Gatt and Krahmer 2018], when application domains are small and variation is expected to be minimal, sentence generation is a relatively straightforward task, and outputs can be specified using templates.

For example, the template `Turn [direction] onto [road] and continue for [distance] meters` is a meta-sentence with three placeholders, indicated by brackets, that can be filled with a direction, a thoroughfare indication, and the distance one must keep producing for example `Turn right onto 5th Avenue and continue for 200 meters`.

An advantage of templates is that they are easy to implement and they ensure a good quality of the output by avoiding the generation of grammatically incorrect text structures. The disadvantage of templates is that they might not scale well to applications that require considerable linguistic variation [Gatt and Krahmer 2018].

2.3.2. Neural-based Generation

Neural-based generation uses language models to generate natural, meaningful phrases and sentences. These models are used to generate natural sentences based on neural networks.

The use neural-based generation has the advantage of insert variability to the generated dataset and reduce the effort from the template designer. The designer just set a few examples in the template and, this generated example from a template is used to train the neural models.

3. Experimental Setup and Results

3.1. Datasets

We choose as a domain a collection of frequently asked questions (FAQ) about several public services of the government of the state of Ceará[1]. We intend to simulate part of the task of migrating a search-based portal to a dialog-based information service.

[1]Available at `https://cartadeservicos.ce.gov.br/`

We select six public services and nine classes of frequently asked questions related to them. Each class of questions is related to a possible intent that can be raised in a conversation in natural language. Table 1 shows the set of intents.

ID	**Intent**
servico	to know what a service is about
loc_presencial	to know whether a service requires presence *in loco*
loc_presencial_obj	to know whether a service is available in a given city
documentos	to know which documents are required by a service
doc_obj	to know whether a document is required by a service
doc_estado_civil	to know which documents are required by a service for a given marital status
doc_estado_civil_obj	to know whether a document is required by a service for a given marital status
hor_funcionamento	to know the opening hour of a given service provider
hor_funcionamento_obj	to know whether a service is available in a given time

Table 1. Intents of the public services chatbot domain.

First, we build five synthetic datasets to evaluate the effects of template-based and neural-based data for the initial training of chatbot systems (Table 2). Each dataset entry contains a text in natural language corresponding to a dialogue question, an intention descriptor, and a list of entities presented in it.

TBD1 and TBD2 datasets were generated by a set of hand-coded template questions. Each template question is a text associated with a single intention. Let `audiometria`, `mamografia`, and `radiografia` be terms related to an entity class called `servico` (service), the template `O que é [mamografia](servico)?` produces the grounded entries: `O que é audiometria?`, `O que é mamografia?`, and `O que é radiografia?`.

TBD2 dataset is built based on seven template questions for each intent with up to three placeholders for six different entities class: `servico` (service), `horario` (time), `dia` (date), `local` (local), `documento` (document), and `estado civil` (marital status). Each placeholder is replaced by terms related to its respective entity classes, such as `audiometria` and `mamografia` for `servico`, including some synonyms variation such as `radiografia` and `raio-x`.

Note that a template-based dataset will grow exponentially based on the number of placeholders within its template questions. Thus we limit in 600 the number of

Dataset	Description
TBD1	Template-based small dataset with 4844 entries
TBD2	Template-based large dataset with 5324 entries
TBTD	Template-based test dataset with 1254 entries
P5D1	PTT5-generated small dataset with 3391 entries
P5D2	PTT5-generated large dataset with 3727 entries

Table 2. Generated datasets.

Tokenizer	Intent Classifier
BERT	DIET
BERT Multilingual	DIET
BERTimbau	DIET
BERT Multilingual	BERT Multilingual
BERTimbau	BERTimbau

Table 3. NLU models used in the experiments.

grounded entries generated for each intent, randomizing the choice of terms to replace each placeholder.

In order to investigate how dataset variety influences the quality of the final result, we build the TBD1 dataset intentionally less diverse, with fewer synonyms for each entity and with placeholders happening only at the beginning or end of each template question.

TBTD is a test dataset generated using the same methodology of TBD2 but with no common entries between them. TBTD is meant to be used to test the overall performance of NLU models trained with the other synthetic datasets. We also built two datasets *TBD1small* and *TBD2small* to be used on the neural-based generation with 1453 and 1597 entries, respectively, with no intersection with TBD1 and TBD2.

We build P5D2 by training PTT5 to generate sentences in natural languages giving a pair intent-entities as input and comparing its output with the respective sentence associated in the *TBLSsmall* datasets. After training PTT5, we build P5D2 as the collection of sentences generated when we give the intent and entities presented in TBD2 as input to PTT5. The P5D2 dataset has BLEU of 56.6332, F1 of 0.7273, and exact match of 0.3565. The same process is applied to build P5D1 using *TBD1small* and TBD1 datasets. The P5D1 dataset has BLEU of 62.2126, F1 of 0.7670, and exact match of 0.4638.

3.2. Experimental Setup

We evaluate the use of synthetic data by training five different NLU models. Each model is a combination of a model tokenizer and an intent classifier as indicated in Table 3. Our intent is to emulate five different approaches that can be used in the development of a chatbot for Portuguese.

The first combination BERT+DIET[2] is one of the standard options when developing a chatbot using RASA. We use it as a baseline for our experiments. It is expected that this model does not perform well when dealing with Portuguese sentences. Thus we built two other models combining the DIET classifier with BERT Multilingual [Devlin et al. 2019] and BERTimbau [Souza et al. 2020], respectively.

We build two intent classifiers based on BERT multilingual and BERTimbau model architectures, simulating the case where a developer creates his own component. As before, our intention is also to approximate the intent classifier component to language models trained for Portuguese.

Based on these five models we design the following experiments: (a) each model will be fine-tuned with each one of TBD1, TBD2, P5D1, and P5D2 datasets; (b) we

[2]The RASA's BERT default tokenizer is loaded with `rasa/LaBSE` model weights available in https://huggingface.co/rasa/LaBSE.

		F1			Acuracy			Recall		
Dataset	**NLU Model**	0%	70%	90%	0%	70%	90%	0%	70%	90%
TBD1	BERTimbau	0.988	0.965	0.849	0.988	0.977	0.818	0.988	0.943	0.818
	BERTimbau + DIET	0.949	0.903	0.874	0.949	0.901	0.874	0.949	0.901	0.874
	BERT multilingual	0.902	0.794	0.742	0.905	0.826	0.695	0.905	0.763	0.693
	BERT mult. + DIET	0.931	0.892	0.888	0.931	0.891	0.891	0.931	0.891	0.891
	BERT + DIET	0.881	0.876	0.846	0.883	0.872	0.849	0.881	0.872	0.849
TBD2	BERTimbau	0.993	0.993	0.885	0.997	0.997	0.869	0.998	0.998	0.885
	BERTimbau + DIET	0.999	0.997	0.997	0.999	0.998	0.998	0.998	0.998	0.998
	BERT multilingual	0.998	0.998	0.909	0.997	0.997	0.908	0.997	0.997	0.909
	BERT mult. + DIET	0.999	0.997	0.995	0.999	0.997	0.995	0.998	0.997	0.995
	BERT + DIET	0.997	0.997	0.996	0.998	0.997	0.996	0.997	0.997	0.997
P5D1	BERTimbau	0.998	0.897	0.696	0.998	0.946	0.639	0.998	0.864	0.639
	BERTimbau + DIET	0.861	0.839	0.817	0.873	0.850	0.834	0.873	0.850	0.845
	BERT multilingual	0.996	0.839	0.753	0.996	0.857	0.674	0.996	0.787	0.674
	BERT mult. + DIET	0.810	0.808	0.768	0.816	0.814	0.776	0.816	0.814	0.776
	BERT + DIET	0.750	0.729	0.724	0.758	0.736	0.715	0.759	0.736	0.725
P5D2	BERTimbau	0.971	0.954	0.853	0.972	0.966	0.813	0.972	0.938	0.813
	BERTimbau + DIET	0.916	0.910	0.899	0.915	0.912	0.767	0.921	0.912	0.904
	BERT multilingual	0.959	0.945	0.901	0.959	0.939	0.936	0.959	0.918	0.836
	BERT mult. + DIET	0.908	0.900	0.897	0.909	0.902	0.899	0.909	0.902	0.899
	BERT + DIET	0.840	0.793	0.784	0.847	0.803	0.801	0.846	0.803	0.795

Table 4. Result of the intent classification experiments.

measure accuracy, F1, and recall of predicting intents of the TBTD dataset; (c) these metrics will be evaluated with three confidence scenarios: minimum of 90%, minimum of 70% and no minimum required.

The confidence scenarios aim to emulate a common requirement of dialogue systems where a chatbot should proceed in a certain dialogue flow only when the confidence in the predicted intent exceeds a given threshold.

3.3. Intent Classification Results

Table 4 summarizes the results of our experiments. The overall result is that all generated datasets provide good training data. The experiments show that, with some exceptions, most of the trained models have an F1 score over 0.8 with these datasets.

The best result is obtained with TBD2 where all models show an F1 score of 0.99, except by BERTimbau and BERT Multilingual with 90% of confidence (0.885 and 0.909, respectively). This is an expected result since TBD2 is more structurally similar to the test dataset than the others.

The P5D1 dataset shows the lowest results among the training datasets. In the BERT+DIET experiment, for example, we obtained an F1 score of 0.72 with 90% of confidence (27% less than the TBD2 experiment). However, these models show F1 scores up to 0.99 when we relax the confidence restriction.

By comparing the intermediate results of TBD1 and P5D2, P5D2 seems to provide a better result than TBD1. Although for BERTimbau they show a variation of F1 score of 1% to 3%, for BERT multilingual the difference of F1 score is up to 15% in the experiment with 90% of confidence.

When we compare TBD1 to P5D1 and TBD2 to P5D2, it is possible to notice that the results of template-based datasets are slightly superior to those neural-based generated. This means that T5 was able to extract information from the training data and generate databases similar to those template-based generated, but with small errors in some texts, reducing the models' performance. The greater amount of entries in the template-based dataset could also be positively influencing the overall result.

Considering what was observed in these experiments, generating datasets from text templates seems a good choice when a real database is not available and you have an expert available for the task. One main concern is that it can lead to the generation of bases with many similar texts, limiting the model's ability to correctly classify texts that are very different from the training dataset On the other hand, datasets generated with T5 seem to be a good option when there is at least a small initial dataset to seed the neural-based generation, considering that this approach still achieves good general results.

4. Conclusions

In this work, we evaluate the performance of methods to generate synthetic data for chatbots in the domain of public services' FAQ provided by the government of the state of Ceará. We evaluate two methodologies to generate synthetic datasets: one based on template generation and the other based on neural generation using transformer networks. These datasets were used to train chatbot components in the intent classification task.

We generated two datasets, TBD1 and TBD2, from a set of hand-coded templates and two neural-generated datasets, P5D1 and P5D2, by training PTT5 to generate sentences in natural languages. We then train and evaluate five different models in the task of intent classification.

The best result is obtained with TBD2 while the P5D1 has the lowest. This was an expected result since TBD2 is indeed the most structurally similar to the test dataset and P5D1 besides being the smallest among them also presents small grammatical text errors due to neural-based generation. By comparing the intermediate results of TBD1 and P5D2, P5D2 seems to provide a better result than TBD1.

When we compare TBD1 to P5D1 and TBD2 to P5D2, it is possible to notice that the results of template-based datasets are slightly superior to those neural-based generated, however, datasets generated with T5 seem to be a good option when one has limited access to domain experts.

For future work, we intend to refine the quality of the generated datasets and expand the generation methods to include generation based on structured knowledge, such as an ontology-based generation approach.

Acknowledgments. This work is partially supported by the FUNCAP projects 04772314/2020.

References

Al-Sinani, A. H. and Al-Saidi, B. S. (2019). A survey of chatbot creation tools for non-coder. *Journal of Student Research*.

Amin-Nejad, A., Ive, J., and Velupillai, S. (2020). Exploring transformer text generation for medical dataset augmentation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4699–4708.

Bird, J. J., Ekárt, A., and Faria, D. R. (2020). Chatbot interaction with artificial intelligence: Human data augmentation with T5 and language transformer ensemble for text classification. *arXiv preprint arXiv:2010.05990*.

Bocklisch, T., Faulkner, J., Pawlowski, N., and Nichol, A. (2017). Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.

Bunk, T., Varshneya, D., Vlasov, V., and Nichol, A. (2020). Diet: Lightweight language understanding for dialogue systems. *arXiv preprint arXiv:2004.09936*.

Carmo, D., Piau, M., Campiotti, I., Nogueira, R., and Lotufo, R. (2020). PTT5: Pretraining and validating the T5 model on brazilian portuguese data. *arXiv preprint arXiv:2008.09144*.

Chen, H., Liu, X., Yin, D., and Tang, J. (2017). A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.

Dauphin, Y. N., Tur, G., Hakkani-Tur, D., and Heck, L. (2013). Zero-shot learning for semantic utterance classification. *arXiv preprint arXiv:1401.0509*.

Deng, L., Tur, G., He, X., and Hakkani-Tur, D. (2012). Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 210–215. IEEE.

Deriu, J., Rodrigo, A., Otegi, A., Echegoyen, G., Rosset, S., Agirre, E., and Cieliebak, M. (2021). Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54(1):755–810.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Gatt, A. and Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.*, 61(1):65–170.

Hashemi, H. B., Asiaee, A., and Kraft, R. (2016). Query intent detection using convolutional neural networks. In *International Conference on Web Search and Data Mining, Workshop on Query Understanding*.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Ive, J., Viani, N., Kam, J., Yin, L., Verma, S., Puntis, S., Cardinal, R. N., Roberts, A., Stewart, R., and Velupillai, S. (2020). Generation and evaluation of artificial mental health records for natural language processing. *NPJ Digital Medicine*, 3(1):1–9.

Peng, B., Zhu, C., Li, C., Li, X., Li, J., Zeng, M., and Gao, J. (2020). Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182, Online. Association for Computational Linguistics.

Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. (2014). Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd international conference on world wide web*, pages 373–374.

Souza, F., Nogueira, R., and Lotufo, R. (2020). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*.

Tur, G., Deng, L., Hakkani-Tür, D., and He, X. (2012). Towards deeper understanding: Deep convex networks for semantic utterance classification. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5045–5048. IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, Long Beach, Califórnia.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Tackling neural machine translation in low-resource settings: a Portuguese case study

Arthur T. Estrella[1], **João B. O. Souza Filho**[2]

[1] Electrical Engineering Program (PEE/COPPE), Federal University of Rio de Janeiro
PO Box 68504, RJ 21941-972, Brazil

atelles@coppe.ufrj.br, jbfilho@poli.ufrj.br

Abstract. *Neural machine translation (NMT) nowadays requires an increasing amount of data and computational power, so succeeding in this task with limited data and using a single GPU might be challenging. Strategies such as the use of pre-trained word embeddings, subword embeddings, and data augmentation solutions can potentially address some issues faced in low-resource experimental settings, but their impact on the quality of translations is unclear. This work evaluates some of these strategies on two low-resource experiments beyond just reporting BLEU: errors are categorized on the Portuguese-English pair with the help of a translator, considering semantic and syntactic aspects. The BPE subword approach has shown to be the most effective solution, allowing a BLEU increase of 59% p.p. compared to the standard Transformer.*

1. Introduction

Since the rise of the Neural Machine Translation (NMT) branch, many solutions solely focused on surpassing the state-of-the-art, ignoring the associated computational burden. Thus, most models have been progressively adopting deeper architectures, hugely increasing the number of network parameters and, as a result, the dependence on more extensive datasets. The excessive focus on boosting performance regardless of complexity deviated the researchers from a more profound criticism over how such architectures address the translation task, if more cost-effective models can be proposed, and how to better cope with translation errors.

Low-resource NMT domains are defined as practical development scenarios wherein the GPU memory and the amount of data available to train some model are limited. Some techniques can potentially help under those circumstances, as the prior initialization of neural network embedding weights [Qi et al. 2018] with pre-trained word embeddings, the production of embeddings at a subword level [Sennrich et al. 2015b] or data augmentation with a monolingual dataset [Sennrich et al. 2015a] (also known as back-translation).

To the best of our knowledge, experimental studies discussing and evaluating the cost-effectiveness of strategies aiming to circumvent the practical issues faced with low-resource domains, especially considering the English-to-Portuguese pair, are missing in the literature. Many previous works only focused on optimizing metrics such as BLEU. Despite its usefulness, this index is limited due to only accounting for matches of a fixed number of n-grams, penalizing correct but different lexical translations.

This work[1] uses Transformers [Vaswani et al. 2017] to experimentally evaluate the impact of strategies such as transfer learning (by the use of pre-trained word embeddings), subword modelling, and data augmentation in the translation quality. It considers only one average size GPU and small to medium sized datasets (low-resource), focusing on the English-to-Portuguese pair. Additionally, a qualitative analysis of the translation errors is derived over a sample of sentences by a native translator, considering a multi-dimensional criterion, aiming to evaluate models' performance on a broader scope than BLEU.

This paper is structured as follows: Section II provides a brief coverage of the Transformer architecture, and Section III discusses the main issues to be tackled in low-resource domains, along with some potential strategies that can be exploited in such cases. Section IV provides a brief description of the datasets considered in this work and depicts quantitative and qualitative results for the strategies here considered. Finally, Section VI poses the conclusions and next steps.

2. The Transformer Architecture

Transformers refers to a branch of algorithms based on the seminal work of [Vaswani et al. 2017], representing the state-of-art. Differently from the sequential processing inherent to the Recurrent Neural Networks (RNN) adopted in previous NMT models, the Transformer model processes large sentences in parallel, establishing a richer set of interrelations between source and target sentence words, thus leading to a better inference of the words' context and a higher performance with long sentences. The reader is referred to the original paper for more details about this architecture.

3. Tackling low-resource settings

Low-resource constraints refer to limitations on dataset quality/size and computational resources available. Small datasets can be strongly biased in specific contexts, which may induce the predictions produced by the decoder model to move away from the reference or even turn the training process unstable, reducing the final model performance. In turn, computational aspects are primarily related to the number of GPUs available and their standalone memory. Memory constraints directly affect the definition of training and model hyperparameters, such as the batch size, the number of hidden layers, the embedding size, and the size of the attention mechanisms. To avoid an experiment failure due to an out of memory error, one should first consider the largest sentence size, a common batch size limiting factor. Too small batches may lead to unstable and biased training, increasing the epoch time and resulting in sub-optimal translation quality. Moreover, the estimation a priori of a minimum quantity of sentences for an adequate translation is also a challenging task, severely depending on the complexity of the application domain. Hopefully, the following strategies may mitigate the need for abundant data and GPU memory in the translation task:

3.1. Transfer learning methods

Transfer learning exploits other application parameters as initial values for the NMT model training (warm-start). A typical example refers to using pre-trained word embed-

[1]ACKNOWLEDGMENT - This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil(CAPES) – Finance Code 001.

dings in the embedding layer of neural models instead of the default random parameters' initialization. This procedure accelerates and improves training since these embeddings already carry out some semantic word meanings to be subsequently refined to a particular translation task (fine-tuning).

The use of pre-trained words can be pretty effective in low-resource scenarios, as pointed in the analysis conducted in [Qi et al. 2018]. According to the authors, there is a "sweet spot" for dataset size, according to which this strategy is more effective. Similarly, the impact over more related language pairs is often higher, such as Spanish and Portuguese.

3.2. Subword methods

Subword embeddings represent a useful strategy to reduce the out of vocabulary (OOV) occurrences. The central idea is decomposing words into sub-parts (character groups), which are common to many words, turning the model less susceptible to the vocabulary content and its size. This technique, referred to as BPE (Byte Pair Encoding) - a compression algorithm, was introduced by [Sennrich et al. 2015b]. Roughly, BPE brokes the words in a corpus into smaller parts (the small BPE unit is a character); some of them subsequently merged, with the number of merge operations being the main hyperparameter to be tuned. The BPE drawback is the impossibility of defining a maximum vocabulary size a priori. A clear advantage of moving from word-level to subword NMT using BPE is reported in [Sennrich and Zhang 2019]: an increase of BLEU score from 7.2 to 16.6 in an ultra-low-resource setting as well as a consistent rise in the BLEU values for a wide range of application scenarios.

Pre-trained word embeddings also followed the subword trend with the proposition of the Fast Text algorithm [Bojanowski et al. 2016]. This technique treats words as a bag of character n-grams and adds tokens to distinguish among prefixes, suffixes, and other character sequences. In practice, each word is assigned to a given number of n-grams, typically $3 \leq n < 6$. Finally, the word is represented by the sum of the vector representations (embeddings) associated with the n-grams composing it.

3.3. Data Augmentation

Large-scale parallel corpora is not a common resource for most existing language pairs, unlike monolingual corpora. However, is it possible to exploit the abundant and large monolingual datasets widely available these days for data augmentation? The answer is yes and this technique is referred to as Back-Translation (BT) [Sennrich et al. 2015a]. The idea is quite simple: let us consider the development of a translation model from a language A to B. New pairs of sentences can be simply synthesized by training an inverse model, i.e., a translator from the language B to A, with the same sentence pairs. Once completed this training, this auxiliary model can be fed with corpora from a similar or a different context domain to produce new sentence pairs. BT has shown to be a simple but effective method to address low data availability in many domains, as shown in [Poncelas et al. 2018], often increasing translation performance.

4. Experimental Setup

Datasets with low to medium complexity levels were carefully picked for the proposed analysis: Tatoeba [Tiedemann 2020] and TED Talks [Cettolo et al. 2012]. Both repre-

sent a low-resource scenario due to the scarce number of sentences, in alignment with some references [Sennrich and Zhang 2019] [Zoph et al. 2016]. News Commentary v16 [Tiedemann 2012] is a different domain monolingual dataset used for BT and includes a rich range of sentences in terms of content and complexity.

The reduced Tatoeba dataset contains 143.8k small and basic to intermediate English level sentences, posing a low complexity challenge for the NMT task. It includes 26.3k unique words in PT and 15.3k in EN. TED Talks is a medium-size database covering a range of subjects, including from low to high complex sentences.

In the experiments, 10% of Tatoeba was held out for testing, while the remaining data was split into 10% for validation and 90% for training, using a seed equal 0. Despite TED disposing predefined training, test, and validation sets, the original validation set is too small (906 sentences), leading us to move the last 20 talks (2081 sentences) from the training set to this set. As a result, the training set contains 236.1k (1918 talks) sentences, randomly sampled to defining training batches using a seed equal to 157, and the test set includes 11.4k sentences. Additionally, all text was pre-processed to eliminate all XML enclosed sentences and tags, except for the ones related to title and description.

The experiments were performed on a single GPU, using Google Colaboratory and Kaggle infrastructure. Typically such environments dispose of NVIDIA GPUs like Tesla P100, Tesla K80 or Tesla T4, with a GPU memory ranging from 12GB to 16GB.

5. Results

Regarding the Transformers, the parameters adopted were $d_{model} = 256$, $d_{ff} = 256$, 8 attention heads, and Q, K and V square matrices with dimension 64. The pre-trained Fast Text-based models, which employed embeddings described in [Hartmann et al. 2017], are exceptions, considering $d_{model} = 300$ and 6 attention heads. All variants adopted the Adam optimizer with $\beta \in (0.9, 0.98)$ and $\epsilon = 10^{-8}$, a learning rate of 10^{-4} and the beam search considered a beam with size 3. The early stopping criterion was based on the validation perplexity behaviour for ten epochs, halting the training in case of performance stagnation.

Sacrebleu [Post 2018] and NLTK [Loper and Bird 2002] are two BLEU variants used for assessing the performance of the models. The major difference between them resides in a stronger Sacrebleu's penalization over cases where the translated and reference sentences differ in length.

5.1. Effects of restricting dataset content

To shed light on the possible effects of limited data on the performance of NMT models, we considered a hypothetical experimental scenario where only a fraction of TED and Tatoeba training sets were used in training, according to the following percentages: 33.3%, 50%, 66.6%, 83.3%. Table 1 summarizes the results.

Results show that the Sacrebleu scores for Tatoeba were about twice the achieved with TED, corroborating with the much higher complexity of the latter. The BLEU metrics for both datasets have shown a monotonic behaviour, with exceptions to TED in two cases: Sacrebleu (100% $\times$ 83.3%) and NLTK (100% $\times$ 83.3 and 83.3% $\times$ 66.6 %). The reasons for such findings may include: (1) the possible use of synonyms in the translations, an aspect ignored by any BLEU metric; (2) a higher incidence of repetition errors

Table 1. Data augmentation scores

Fraction of the Dataset	Tatoeba				TED			
	Sacrebleu	NLTK BLEU	Batch Size	Epochs	Sacrebleu	NLTK BLEU	Batch Size	Epochs
33.3%	48.64	67.09	512	76	24.7	57.65	30	40
50%	52.53	70.12	512	65	25.18	56.46	30	40
66.6%	55.3	72.12	512	58	26.22	**56.81**	29	36
83.3%	56.24	73.18	512	58	**26.74**	56.57	28	30
100%	**57.99**	**74.07**	512	58	25.24	55.36	28	30

due to data quality issues (to be discussed further in the following section); (3) the more complex and richer TED content, which might have led to a wider subject coverage in the training set, reducing model accuracy, a hypothesis deserving a future investigation. Finally, models developed with a fraction of the original training datasets (66.6%) performed surprisingly well.

5.2. Effects of transfer learning and subword embeddings strategies

Aiming to evaluate the leveraging effects of pre-trained Fast Text and BPE [Sennrich et al. 2015b] strategies in low-resource NMT tasks, BPE models were implemented in the Texar framework [Hu et al. 2019] (PyTorch version). In contrast, the alternative models considered a customized PyTorch [Paszke et al. 2019] solution. Table 2 exhibits these results, reproducing the last line of Table 1 to allow an easier comparison of the results.

Table 2. Transfer learning and subword embeddings translation results

Technique applied	Tatoeba				TED			
	Sacrebleu	NLTK BLEU	Batch Size	Epochs	Sacrebleu	NLTK BLEU	Batch Size	Epochs
None	57.99	74.07	512	58	25.24	55.36	28	30
Fast text	56.96	69.91	512	50	24.07	51.69	30	45
Subword BPE	**66.63**	**83.02**	512	40	**40.26**	**72.20**	32	40

Curiously, the use of Fast Text embeddings is associated with an unexpected performance drop for both datasets. Conversely, the gains observed with BPE, which also exploits word embeddings, were impressive. One hypothesis for the bad Fast Text performance is a possible overspecialization to other text domains, since it was produced with content mined by a crawler [Hartmann et al. 2017]. The higher BPE gain in TED (15.02) compared to Tatoeba (8.64) signalizes the effectiveness of BPE in dealing with more complex NMT scenarios, especially regarding a more diverse vocabulary, avoiding OOV occurrences.

5.3. Effects of the Back-Translation (BT) strategy

The BT experiments were restricted to the TED dataset. Data augmentation was performed with synthetic sentences produced with the own TED (using its left out sentences) and with the News dataset. These experiments aimed to verify if data augmentation can result in higher BLEU scores under low-resource constraints.

A single EN-PT Transformer was trained with the entire TED dataset to generate the synthetic sentences, reaching 27.73 and 63.8 points for the Sacrebleu and NLTK, respectively. The subset of back-translated sequences appended to the training sets was randomly sampled using the following seeds: 157 (TED) and 0 (News).

Table 3. TED Talks back-translation results

Technique applied	Batch size	Epochs Trained	Sacrebleu	NLTK BLEU
None (Original TED)	30	27	25.24	55.36
Reduction of TED to 50%	40	30	25.18	**56.46**
BT (50% of News synthetic examples)	34	33	21.80	51.34
BT (50% of TED synthetic examples)	34	28	**25.95**	56.42
Reduction of TED to 66.6%	36	29	26.22	56.81
BT (33.3% of News synthetic examples)	34	27	24.12	53.77
BT (33.3% of TED synthetic examples)	34	27	**27.54**	**58.95**
Reduction of TED to 83.3%	28	30	26.74	56.57
BT (16.6% of News synthetic examples)	34	29	31.28	63.30
BT (16.6% of TED synthetic examples)	34	27	**34.62**	**64.61**

Table 3 exhibits the results. For a more severe restriction on the dataset size (50%), using other domain synthesized sentences is harmful to model performance, while own-domain synthesis resulted in a marginally better BLEU score. However, for a lower percentage of synthetic data, positive effects start to appear. Considering an intermediate restriction ($\approx 33\%$), using the same domain sentences in back-translation led to a mild increase in both BLEU values compared to the Original TED, signalizing that such "noisy" sentences may contribute to increasing translation quality. Finally, considering a small restriction ($\approx 16.6\%$), both domain approaches are quite effective, resulting in models that largely surpasses the model developed over original data.

5.4. Subjective evaluation

This analysis focused on two dimensions: sentence complexity and error patterns. Random samples were selected from TED, analysed by a human translator, and stratified according to the CEFR scale [Council]. Due to dataset characteristics, this study was restricted to sentences classified as A1, A2, B1 and B2. Ten sentences from each level were presented to two models: the Transformer trained over a fraction of 66% of original data and the BPE variant developed over the entire dataset. The idea here was twofold: first, evaluate the effects of restricting the dataset size over the error patterns; and second, assessing qualitatively the translations produced by the model of best performance, and thus the impact of eliminating <unk> occurrences, generating custom words, and switching from word to subword level.

Regarding the identification of error patterns, a multidimensional evaluation in eight categories was considered: similar word choice, omission, out of context, verb tense, sentence choice (the translation is OK, but the outcome is entirely different from the reference), insertion, repetition and <unk> errors [2]. Table 4 shows the number of errors committed by each model, stratified by sentence complexity and error category. Considering the limitations of such analysis, such as the reduced sample and the analysis of

[2] A detailed error description and some evaluation samples can be found at `https://github.com/Art31/pt-nmt-low-resource.git`.

only one translator, both models performed quite similarly regarding the ”similar word choice” occurrence. Nonetheless, the BPE produced fewer errors related to ”omission” (levels A1 and B1), ”sentence choice” (B1), ”insertion” (A1, A2 e B1), ”repetitions” (all), and ”<unk> errors” (all), performing worse regarding ”out of context” and ”verb tense”.

Table 4. Class-error ratios per dataset and sentence complexity.

Model Name	Complexity	Similar word choice	Omission	Out of context	Verb tense	Sentence choice	Insertion	Repetition	<unk> error
66% TED	A1	2/10	3/10	0/10	0/10	1/10	**6/10**	3/10	2/10
	A2	7/10	6/10	1/10	1/10	3/10	4/10	2/10	2/10
	B1	7/10	5/10	**3/10**	3/10	3/10	2/10	2/10	3/10
	B2	**8/10**	7/10	**2/10**	**7/10**	**5/10**	5/10	**6/10**	**5/10**
	Average	60.0%	52.5%	15.0%	27.5%	30.0%	**42.5%**	**32.5%**	**30.0%**
BPE	A1	2/10	0/10	0/10	1/10	1/10	2/10	0/10	0/10
	A2	7/10	**6/10**	**3/10**	4/10	**3/10**	1/10	0/10	0/10
	B1	4/10	3/10	1/10	3/10	1/10	0/10	0/10	0/10
	B2	**8/10**	5/10	**3/10**	**5/10**	2/10	**6/10**	**1/10**	0/10
	Average	52.5%	35.0%	17.5%	32.5%	17.5%	**22.5%**	**2.5%**	**0.0%**

Results from Table 4 underwent a Multiple Fisher test to evaluate if the differences observed between the error ratios of the two models are statistically significant. This analysis considered multiple 2x2 tables (one to each class of error), with rows defining the model and columns associated with the occurrence or not of some class of error. The significance level was set to 5%; thus, the null hypothesis was rejected whenever the p-value was lower than 0.05, representing a statistically significant difference. This analysis concluded that the ”repetition error” ($p = 0.0002$), the ”<unk> error” ($p = 0.0001$) and the ”insertion error” ($p = 0.0001$) are indeed less frequent in BPE than 66% TED.

6. Conclusion

This paper focused on dealing with low-resource NMT scenarios, considering low and medium complexity Portuguese-English datasets (TED and Tatoeba). It experimentally evaluated the impact of transfer learning (pre-trained word embeddings), subword embeddings (BPE), and Back-Translation strategies (using the same and different domains data) over BLEU performance. In addition, this work presented a qualitative analysis conducted by a human translator over the outcomes of some best performing models, considering a specifically designed multidimensional evaluation criteria, for a sample constituted by a total of 40 sentences, equally stratified in four CEFR levels.

The BPE was the most effective technique for dealing with a low-resource setting, attaining the highest BLEU values and the lower error rates in six from eight error categories defined by the qualitative analysis. Same domain data augmentation has also led to exciting results when synthesising only a small portion of the original training set (16.6%).

Future works include evaluating models exploiting both BPE and BT and considering more sentences, as well as CEFR levels, in the qualitative analysis, possibly bringing a clearer view of error patterns and enlighting the practical effects of each strategy in objective and subjective translation quality aspects.

References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Cettolo, M., Girardi, C., and Federico, M. (2012). WIT3: Web inventory of transcribed and translated talks. *Proc. of EAMT*, pages 261–268.

Council, E. The CEFR levels - council of europe (coe). `https://tinyurl.com/cefrlcoe`. Accessed: 2021-08-12.

Hartmann, N., Fonseca, E. R., Shulby, C., et al. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *CoRR*, abs/1708.06025.

Hu, Z., Shi, H., Tan, B., et al. (2019). Texar: A modularized, versatile, and extensible toolkit for text generation. In *ACL 2019, System Demonstrations*.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *Proc. of the ACL Workshop on Effective Tools for Teaching Natural Language Processing.*

Paszke, A., Gross, S., Massa, F., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Proc. Systems 32*, pages 8024–8035. Curran Associates, Inc.

Poncelas, A., Shterionov, D. S., Way, A., et al. (2018). Investigating Back-translation in neural machine translation. *CoRR*, abs/1804.06189.

Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proc. of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Assoc. for Comp. Linguistics.

Qi, Y., Sachan, D., Felix, M., et al. (2018). When and why are pre-trained word embeddings useful for neural machine translation? pages 529–535, New Orleans, Louisiana. Assoc. for Comp. Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2015a). Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Sennrich, R., Haddow, B., and Birch, A. (2015b). Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In *Proc. of the 57th Annual Meeting of the Assoc. for Comp. Linguistics*, pages 211–221, Florence, Italy. Assoc. for Comp. Linguistics.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In *Proc. of the Eight International Conf. on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Assoc. (ELRA).

Tiedemann, J. (2020). The Tatoeba translation challenge - realistic data sets for low resource and multilingual MT. *CoRR*, abs/2010.06354.

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Zoph, B., Yuret, D., May, J., and Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proc. of the 2016 Conf. on Empirical Methods in Natural Language*, pages 1568–1575, Austin, Texas. Assoc. for Comp. Linguistics.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Uma revisão breve sobre perguntas complexas em bases de conhecimento para sistemas de perguntas e respostas

João Gomes Jr.[1], **Rômulo Chrispim de Mello**[2], **Ana Beatriz Kapps dos Reis** [2], **Victor Ströele**[2], **Jairo Francisco de Souza**[1,2]

[1] Laboratório de Aplicações e Inovação em Computação (LApIC)
Programa de Pós-Graduação em Ciência da Computação – UFJF

[2]Departamento de Ciência da Computação – UFJF
36.360-900 – Juiz de Fora – MG – Brasil

{joraojunior,romulomello,anabeatrizkapps}@ice.ufjf.br

{victor.stroele,jairo.souza}@ice.ufjf.br

Abstract. *The advance on Question Answering systems has achieved significant results, and new related problems have arisen. Few studies have addressed the Complex Knowledge Base Question Answering (C-KBQA) problem. This work presents an overview of C-KBQA systems. A collection of 54 papers were systematically selected, and a map of the methods, approaches, trends, future directions, and gaps in the C-KBQA research was performed. This study shows that C-KBQA systems need to handle multi-hop and constraint questions, and two approaches are usually used to address this problem.*

Resumo. *O avanço nos sistemas de Question Answering alcançou resultados importantes e novos problemas relacionados, como Complex Question Answering e Knowledge Base Question Answering, surgiram. No entanto, faltam estudos que analisam o problema e abordagens para Complex Knowledge Base Question Answering (C-KBQA). Este trabalho preenche essa lacuna apresentando uma visão geral do C-KBQA. Uma coleção de 54 artigos foi selecionada e um mapa dos métodos, abordagens, tendências e lacunas sobre C-KBQA foi realizado. É mostrado que as questões de múltiplos saltos e restritivas são os dois tipos de questões abordadas na literatura. Três etapas foram identificadas para criar um sistema C-KBQA e duas abordagens são geralmente usadas.*

1. Introdução

Sistemas de Perguntas e Respostas (do inglês *Question Ansewring* - QA) têm o propósito de recuperar a informação mais relevante (resposta) para uma pesquisa (pergunta) feita por um usuário [Croft et al. 2010]. Diferente dos motores de busca, os sistemas de QA visam encontrar as respostas exatas para uma pergunta em linguagem natural (do inglês *Natural Languague Question* - NLQ) [Yin et al. 2014, Rodrigo and Penas 2017]. Para fazer isso, os sistemas de QA precisam reconhecer as informações dentro de uma NLQ. Esta tarefa implica na identificação de objetos relevantes e suas conexões, extraindo as principais descrições ou ideias que estão contidas em uma pergunta. O mapeamento de uma pergunta para seus principais assuntos (conceitos, organizações, pessoas, etc) é uma tarefa que tem sido explorada para sistemas de QA [Hao et al. 2019, Hua et al. 2020a].

Bases de conhecimento (do inglês *Knowledge Base* - KB) é um modelo de dados baseado em uma rede semântica que, usualmente, utiliza um formato de triplas para representar e relacionar as informações contidas em um domínio de dados [Ji et al. 2020]. Os sistemas de QA que fazem uso de KBs são chamados *Knowledge Base Question Answering* (KBQA). Estes sistemas usam essas estruturas semânticas, por exemplo, Freebase ou Wikidata para responder diretamente uma pergunta. Portanto, os sistemas de KBQA extraem os principais recursos do texto e os mapeiam em uma KB para responder a uma pergunta. O uso de KBs fornece um resultado mais preciso e conciso, uma vez que uma NLQ pode ser mapeada para consultas estruturadas em uma KB [Cui et al. 2019].

Mesmo com o uso de KBs, os sistemas de QA ainda precisam lidar com diferentes tipos de perguntas. Pode-se dividir esses tipos de perguntas em dois grupos: simples e complexas. Perguntas simples são aquelas que contêm respostas diretas e apenas um fato precisa ser detectado para obter a resposta [Bordes et al. 2015]. Já perguntas complexas precisam de mais informações do que as explícitas que podem ser extraídas diretamente. Nesse caso, é necessário realizar consultas avançadas para coletar a resposta dos KBs, como a exploração de relações indiretas entre entidades, multi-relações, restrições qualitativas e quantitativas, entre outras [Bao et al. 2016]. Atualmente, os sistemas de QA obtêm melhores resultados ao responder a perguntas simples e, por conta disso, os sistemas de QA para perguntas complexas estão recebendo atenção [Ding et al. 2019, Bhutani et al. 2020].

Assim, este trabalho visa apresentar uma visão geral de sistemas *Complex Knowledge Base Question Ansering* (C-KBQA). A compreensão das soluções para C-KBQA inclui a investigação das técnicas mais utilizadas, as principais soluções atuais, onde essas soluções são aplicadas, e a identificação dos principais desafios desse campo de estudo. As principais contribuições deste trabalho são: (i) uma visão geral do estado da arte em C-KBQA; (ii) uma coleção de 54 artigos selecionados sistematicamente a partir de 898 artigos; (iii) um mapeamento de métodos e abordagens usadas no cenário de perguntas e respostas complexas. É mostrado que os sistemas de C-KBQA tentam resolver dois tipos de perguntas complexas: perguntas com múltiplos saltos e perguntas restritivas. Além disso, existem três etapas para construir sistemas de C-KBQA e duas abordagens são usadas nesse processo. Por fim, são discutidos os desafios e tendencias para C-KBQA.

Este trabalho é assim estruturado: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve o protocolo do mapeamento sistemático. A Seção 4 apresenta as discussões do mapeamento. A Seção 5 apresenta as tendências e desafios futuros pra os sistemas de C-KBQA. A Seção 6 apresenta as considerações finais do trabalho.

2. Trabalhos Relacionados

De acordo com [Dimitrakis et al. 2019], o problema de responder perguntas complexas foi reconhecido como um desafio para sistema de QA sobre dados ligados em [Höffner et al. 2017, Rodrigo and Penas 2017]. No entanto, faltam trabalhos que apresentem um mapeamento para esse problema no campo KBQA. Trabalhos anteriores focam em sistemas de perguntas e respostas simples e apresentaram uma breve introdução ao assunto de perguntas complexas, mesmo em revisões recentes de QA [Höffner et al. 2017, Wu et al. 2019b, da Silva et al. 2020]. Esses trabalhos não deixaram claro o problema do C-KBQA, quando surgiu e como ele vem sendo resolvido na

literatura.

Em [Höffner et al. 2017], os autores realizaram um levantamento sobre os desafios de responder a perguntas na Web semântica. Os autores reúnem um total de 62 sistemas desenvolvidos entre 2010 e 2015. No entanto, o problema da pergunta complexa não é discutido em detalhes. Os autores têm apenas uma seção que apresenta uma visão geral de alto nível do problema de responder a perguntas complexas. Isso era esperado uma vez que os artigos analisados pelos autores foram publicados até o início de 2015 e o assunto de pergunta complexa ainda não estava profundamente enunciada para ser pesquisada. [Höffner et al. 2017, Rodrigo and Penas 2017] discutem que os sistemas ainda não estão preparados para ter um bom desempenho dentro do cenário de perguntas e respostas complexas como já fazem no cenário de perguntas simples. No entanto, com o crescimento das pesquisas na área de perguntas complexas, fica evidente o contrário.

Além disso, em [Dimitrakis et al. 2019] os autores apresentam um mapeamento geral do cenário KBQA. No entanto, eles discutem apenas pequenas lacunas no cenário de perguntas complexas mostrando também que as pesquisas em perguntas complexas precisam de melhorias. Isso confirma a necessidade de estudar e compreender este problema. O principal objetivo deste trabalho é ajudar outros pesquisadores(as) da área e permitir que eles(as) entendam os principais conceitos e desafios do C-KBQA.

Este trabalho se diferencia dos demais por realizar uma análise dos métodos propostos para os sistemas de C-KBQA. Além disso, realizamos uma abordagem sistemática com o objetivo de encontrar os trabalhos de forma mais precisa e imparcial, dando a outros pesquisadores a opção de reproduzir os processos que foram feitos. O trabalho apresenta um mapeamento quantitativo e qualitativo de diversos aspectos do CKBQA e permite uma visão geral do que tem sido feito na área. Portanto, este estudo facilita o primeiro contato de novos pesquisadores com o tema.

3. Mapeamento Sistemático

O processo utilizado neste trabalho foi baseado no protocolo apresentado por [Neiva et al. 2016]. O processo consiste nas seguintes etapas: (i) definição das questões de pesquisa, (ii) seleção dos termos de pesquisa relevantes, (iii) definição dos critérios de exclusão e (iv) seleção dos repositórios de pesquisa.

As questões de pesquisa objetivam categorizar e criar uma visão geral da literatura, descobrindo tópicos cobertos na área de pesquisa [Petersen et al. 2015]. Foram definidas duas questões de pesquisa (QP): “Que tipos de perguntas são definidas como perguntas complexas para os sistemas de C-KBQA na literatura e como elas tem sido resolvidas?”(QP1) e “Quais são os recursos e métodos usados para resolver o problema C-KBQA?”(QP2). O método PICOC (*Population*, *Intervention*, *Comparison*, *Outcome*, e *Context*) foi usado para definir o escopo da pesquisa [Petticrew and Roberts 2006]. A Tabela 1 descreve os elementos PICOC e os termos de pesquisa para cada elemento. A *string* de busca foi criada usando os termos de pesquisa, onde cada elemento foi separado por um *AND* e seus sinônimos foram separados por *OR*.

Foram definidos 6 critérios de exclusão (CE) para que trabalhos não relacionados fossem excluídos da avaliação: Duplicado (CE1); Não apresenta um sistema de KBQA (CE2); Não trata o problema de pergunta complexa (CE3); Não escrito em inglês (CE4);

Tabela 1. Definição do PICOC e os termos de pesquisa.

Elemento	Descrição	Termos e Sinônimos
Population (P)	Artigos que apresentam sistemas de QA	*question answering, qa, semantic search, search engine, answering engine*
Intervention (I)	Abordagens para resolver perguntas complexas	*complex question, complex information, complex queries, complex query, complex answer*
Comparison (C)	-	-
Outcome (O)	As soluções para responder a perguntas complexas	*method, technique, algorithm, approach, application, system*
Context (O)	Sistemas de QA que fazem uso de bases de conhecimento	*knowledge base, knowledge graph, kb, kg, linked data, linked open data, lod, semantic web, semantic data*

Literatura cinzenta[1] (CE5); Indisponíveis na íntegra (CE6). As bases *Scopus*, *Google Scholar*, *ISI Web of Science*, *IEEE Digital Library* foram selecionados para a execução da *string* de busca. A coleta dos artigos foi finalizada em 17 de novembro de 2020 e 898 artigos foram obtidos. Os artigos duplicados foram excluídos e a leitura do título e do resumo foi realizada. Os artigos restantes tiveram suas introduções e conclusões lidas. Foram selecionados 45 artigos por responderam as QP. Finalmente, 9 artigos foram adicionados após a etapa de *snowballing*, resultando nos 54 artigos mapeados. Esta etapa tem como objetivo encontrar artigos relevantes que não foram retornados pela *string* de busca, observando os trabalhos citados nas referências dos artigos aceitos.

4. Relatório do mapeamento sistemático

Esta seção discute os dados coletados para responder às perguntas da pesquisa. Os artigos analisados estão disponíveis e informações adicionais podem ser encontradas aqui[2].

4.1. Resultados relativos a QP1

Foram identificados dois subtipos de perguntas complexas: perguntas com múltiplos saltos e perguntas com restrições. Foram encontrados 52 trabalhos abordando perguntas com múltiplos saltos e 17 trabalhos abordando perguntas com restrições. Alguns trabalhos tentam resolver os dois tipos de perguntas complexas simultaneamente e que o termo "perguntas com multi-restrições" se refere a união das categorias listadas anteriormente. Ao abordar perguntas com múltiplos saltos, um sistema C-KBQA tem que lidar com várias entidades que podem ser extraídas de uma pergunta. As entidades detectadas nessas perguntas precisam ser vinculadas e tratar relações indiretas, ao contrário de perguntas simples que podem ser respondidas diretamente. As triplas (sujeito, predicado, objeto) são exploradas dentro de uma KB, e os sistemas fazem saltos entre os objetos detectados na pergunta e as entidades/predicados do KB para obter a resposta. Quando relacionado a questões com restrições, a pergunta inclui algumas restrições que limitam as opções de resposta para uma determinada pergunta [Shin and Lee 2020]. Essas restrições podem ser de vários tipos, por exemplo, temporal ("... antes de 2000"), ordinal ("O primeiro que ..."), quantitativa ("... tendo mais de 5 ..."), dentre outras. Essas restrições podem modificar o assunto principal e consequentemente, alterar a resposta a ser obtida.

[1]Artigos sem revisão dos pares, como, por exemplo, *pre-prints*, relatórios técnicos, patentes, etc.

[2]A lista de artigos e informações adicionais podem ser acessadas em `https://github.com/lapic-ufjf/CKBQA-systematic-mapping-2021`

Os artigos da categoria múltiplos saltos tentam resolver este problema detectando as entidades e relações, criando uma lista de possíveis candidatos para realizar os saltos entre as relações e predicados. Para as perguntas com restrições, a criação de um modelo de perguntas, regras de restrições e decomposição de perguntas é o caminho mais utilizado. O custo computacional é um dos principais problemas que artigos procuram resolver em C-KBQA. Primeiro, uma pergunta pode precisar de muitos saltos para obter a resposta. Em segundo lugar, os sistemas de C-KBQA podem gerar um alto nível de lista de candidatos, uma vez que é necessário certo tempo para processar várias conexões de triplas do KB. Finalmente, uma vez que os dois foram resolvidos, é necessário limitar a lista de candidatos apenas dentro das restrições. A maioria dos artigos tem um módulo apenas para podar e classificar os melhores candidatos para esses problemas.

4.2. Resultados relativos a QP2

Em geral, pode-se dividir o pipeline C-KBQA em três etapas: Análise da pergunta, Representação da pergunta e Classificação de candidatos. Na etapa de análise da pergunta, é realizada a seleção do tipo de pergunta e a identificação dos principais temas. Primeiro, é preciso encontrar os tipos de perguntas que um NLQ possa corresponder, como “quando”, ”o quê”, “como”, entre outros. Esses tipos de perguntas são chamados de *wh-questions*. A marcação Part-of-speech (POS) e as árvores de dependência são geralmente usadas para extrair a semântica da frase e para entender qual é tipo da pergunta. Além disso, o Reconhecimento de Entidade Nomeada e os métodos de reconhecimento de relação são realizados para extrair as principais informações da frase. O DBpedia Spotlight [Mendes et al. 2011], S-MART [Yang and Chang 2015] e Stanford Named Entity Recognizer[3] são exemplos de ferramentas usadas nesta etapa.

Na etapa de representação da pergunta é realizado o mapeamento semântico. Após a análise da pergunta, o sistema C-KBQA possui as entidades e relações extraídas de uma NLQ. No entanto, é necessário mapear e conectar as entidades e relações identificadas para corresponder à estrutura da KB. As pesquisas seguem dois caminhos: abordagens baseadas em análise semântica e abordagens baseadas em redes neurais. Abordagens baseadas em análise semântica (ou baseadas em regras) mapeiam as questões e as informações extraídas em um conjunto de formulários lógicos para serem posteriormente transformadas em modelos de consulta de tripla do KB. Abordagens baseadas em rede neural (ou livre de regras) usam redes neurais para identificar automaticamente os tipos de perguntas e quais são os padrões de consulta mais apropriados para obter a resposta. Ambas as abordagens podem criar um conjunto de candidatos que podem ser considerados como a resposta final. A Seção 4.2.1 apresenta os detalhes sobre essas duas abordagens.

Após a representação semântica e a geração dos candidatos, é realizada a classificação dos candidatos. O objetivo desta etapa é remover as respostas incorretas com base no tipo e na semântica da NLQ original. O melhor candidato é selecionado com base em uma função de avaliação. Para isso, alguns trabalhos utilizam métricas de avaliação como similaridade por cosseno ou função de log-verossimilhança. Em outros casos, um modelo de aprendizado de máquina (ex. SVM) é treinado para coletar os padrões e classificar os candidatos em uma lista dos melhores resultados.

Por fim, as etapas acima são realizadas em duas fases: experimentos *offline* e

[3]`https://nlp.stanford.edu/software/CRF-NER.html`

experimentos *online*. Os experimentos *offline* têm como foco a geração de materiais a serem utilizados nos experimentos *online* e não possuem interação com os usuários. A etapa de criação do candidato é criada na etapa *offline*, pois precisa predefinir o conjunto de regras ou treinar um modelo de rede neural. Os modelos de aprendizado de máquina também precisam ser treinados na etapa *offline* para serem usados na etapa de classificação de candidatos. Nos experimentos *online*, todas as etapas do pipeline são executadas para obter a resposta, utilizando os modelos gerados na etapa *offline*.

4.2.1. Representação das perguntas e geração de candidatos

As abordagens de representação de perguntas e geração de candidatos para C-KBQA podem ser divididas em dois tipos: abordagens baseadas em análise semântica e baseadas em redes neurais. Foram encontrados 47 trabalhos abordando abordagens baseadas em análise semântica e 31 trabalhos abordando abordagens baseadas em redes neurais.

A abordagem baseada em análise semântica (baseada em regras) é o mapeamento de uma NLQ para sua representação de significado [Tong et al. 2019]. A NLQ é transformada em uma representação intermediária que pode ser posteriormente representada como uma forma lógica [Trivedi et al. 2017]. As abordagens nesta categoria criam uma lista de regras predefinidas como expressão lógica, *tempalte questions*, aproximação de sub-grafos, regras gramaticais ou outra estrutura semântica equivalente que representa a semântica da NLQ [Wu et al. 2019a]. Quando a NLQ corresponde a esses padrões, é mais fácil consultá-lo em uma KB. Sistemas de C-KBQA fazem uso da análise semântica para mapear os assuntos da NLQ em dados semanticamente estruturados, mapear os dados estruturados dentro da representação tripla KB e, finalmente, responder à NLQ de forma concisa. Inicialmente, é realizada a quebra da NLQ em sub perguntas. Após a extração do assunto principal da NLQ, cria-se a representação intermediária de cada sub pergunta. Em seguida, a união das formas lógicas de cada NLQ na representação da KB. Por fim, a tripla da KB é criada e pode ser executada posteriormente no esquema da KB.

Esse mapeamento facilita o processo de extração dos termos relevantes de uma pergunta e o processo de conexão com as triplas do KB. No entanto, por ser dependente de regra, às vezes essas abordagens podem não funcionar bem para NLQ que não tem um formato de decomposição e recomposição no conjunto de regras. Essas abordagens também podem ter problemas com a escalabilidade da consulta ao usar muitos relacionamentos, devido a grande quantidade de regras candidatas que, consequentemente, aumentam o número de triplas do KB [Agarwal et al. 2019].

Abordagens baseadas em redes neurais (livres de regras) tentam usar arquiteturas de redes neurais para codificar perguntas e respostas em um modelo de espaço vetorial e selecionar os padrões de consulta mais apropriados para obter a resposta [Luo et al. 2018]. Com isso, é possível identificar tipos de questões e padrões comuns para responder a um determinado tipo de pergunta [Luo et al. 2018]. Os trabalhos nesta categoria geralmente são compostos de uma camada para coletar os *embeddings* e uma camada de rede neural.

Primeiramente, a camada de *embeddings* é usada para transformar a NLQ em uma sequência de vetores de palavras ou vetores de frases. *Embeddings* de palavras reduzem a complexidade computacional, já que as operações de vetores e matrizes são rápidos

de calcular. Word2vec [Mikolov et al. 2013], GloVe [Pennington et al. 2014] e FastText [Bojanowski et al. 2017] são exemplos de arquiteturas de *embeddings* de palavras usadas na literatura por sistemas de C-KBQA. Em seguida, uma rede neural profunda é usada. Nesta etapa, Redes Neurais Recorrentes (RNNs) é a arquitetura mais comum. A RNN é usada para transmitir uma cadeia de informações históricas por meio de uma sequência de unidades de rede neural. A RNN funciona como uma arquitetura de rede em cadeia e analisa a entrada atual e a saída anterior em cada etapa de tempo durante o processamento de dados sequenciais. Assim, a RNN pode extrair a propagação do contexto de informação de uma NLQ. Além disso, a RNN tem sido usada como arquiteturas de codificação e decodificação (*seq-2-seq*). Nesse processo, uma unidade RNN é usada para codificar a NLQ e outra RNN é usada para coletar as informações históricas da NLQ e decodificá-las na sequência de resposta. *Gated Recurrent Unit* (GRU), *Long Short-Term Memory* (LSTM) e suas variantes são as mais usadas para realizar esta etapa, uma vez que essas RNNs podem lidar melhor com o problema da dissipação do gradiente.

Um dos principais problemas da RNN é a queda no desempenho para sequências mais longas e complexas. Para resolvê-lo, trabalhos recentes utilizam o mecanismo de atenção para enfatizar as partes mais relevantes de uma NLQ e preservar o contexto das sentenças [Bhutani et al. 2019, Ding et al. 2019, Tong et al. 2019, Bhutani et al. 2020]. Embora RNNs sejam amplamente utilizadas, Redes de Memória [Miller et al. 2016, Hao et al. 2019, Saha et al. 2018, Hua et al. 2020b, Hua et al. 2020a] e Redes Neurais Convolucionais [Hu et al. 2018, Bao et al. 2016] podem ser usadas nesta etapa. Independente da rede utilizada, esta etapa de treinamento depende de recursos computacionais e pode levar muito tempo para obter resultados satisfatórios, além de depender de conjuntos de dados suficientemente grandes e diversificados para evitar problemas de *overfitting*. Entretanto, bases de dados para sistemas de C-KBQA ainda são um problema em aberto para o cenário, por ser uma campo de pesquisa recente.

As abordagens de análise semântica baseada em redes neurais tentam resolver perguntas complexas usando uma combinação de análise semântica e arquiteturas de redes neurais e estão se tornando o estado da arte [Luo et al. 2018, Ding et al. 2019]. Essa abordagem consiste em treinar uma rede neural para corresponder a um conjunto de regras de análise semântica, em vez de apenas a resposta final. Assim, o modelo aprende a semântica por trás de um NLQ em vez de apenas aprender os padrões de consulta mais apropriados para obter a resposta. Essas abordagens tendem a ser mais generalistas, pois aprendem o passo a passo para responder uma pergunta.

5. Tendências e desafios futuros

Redes Neurais Recorrentes têm se tornado bastante utilizadas neste campo, porém trabalhos recentes mostram que o uso de modelos pré-treinados e transferência de aprendizagem podem ser uma nova opção para treinar mais rapidamente novas soluções. [Lukovnikov et al. 2019] mostrou que modelos pré-treinados como o BERT obtêm bons resultados para responder a perguntas simples. Os autores afirmam que esses modelos podem ter um impacto maior em sistemas de C-KBQA. Avanços em C-KBQA e Deep learning podem criar arquiteturas que demandam menos memória e têm menor custo de treinamento, permitindo a popularização dos modelos KBQA para domínios específicos.

Os sistemas de KBQA são dependentes das informações contidas nas KB e não

é trivial criar sistemas que utilizem informações atualizadas em suas respostas, uma vez que o sistema aprende a responder às perguntas relacionadas ao conjunto de dados de treinamento e teste. É necessário comparar esses sistemas com usuários reais e também avaliá-los em cenários reais. Uma opção para fazer isso é testá-los usando plataformas de *crowdsourcing* e avaliar sua usabilidade. Os sistemas podem fazer uso de abordagens *cross-language* aplicadas ao KB [Schumacher et al. 2020], desambiguação [Kartsaklis et al. 2018], busca em grafos [Namaki et al. 2017], *embedding* [Dettmers et al. 2018] e *reasoning* [Chen et al. 2020]. Além disso, novas técnicas de compreensão de linguagem natural podem melhorar sistemas de perguntas e respostas, como avanços no tratamento de perguntas com ruído [Zhang et al. 2018], novos analisadores de dependência e métodos de rotulagem de função semântica [Zheng and Kordjamshidi 2020], e raciocínio de senso comum [Liu et al. 2020].

Por fim, esses sistemas podem ser frágeis e espúrios. Frágeis porque ainda não são robustos o suficiente e podem falhar em responder uma pergunta quando apenas algumas partes da pergunta são um pouco modificadas, mesmo se o significado principal for preservado [Jia and Liang 2017]. Além disso, são espúrios porque, mesmo se a mesma arquitetura for treinada várias vezes no mesmo conjunto de dados, cada modelo pode memorizar artefatos e vieses diferentes em vez de realmente aprender e, portanto, falhar no desempenho de generalização [McCoy et al. 2019]. É necessário lidar cuidadosamente com a etapa de pré-processamento do conjunto de dados, separá-la com classes de treinamento balanceadas e também tentar explorar algumas paráfrases de perguntas para garantir que a maioria dos tipos de perguntas está sendo explorada.

6. Considerações finais

Este artigo apresentou um mapeamento sistemático sobre *Complex Knowledge Base Question Answering Systems* (C-KBQA). Sistemas de C-KBQA tentam lidar com dois tipos de perguntas complexas: perguntas com múltiplos saltos e perguntas com restrições.

Além disso, foi apresentada uma visão geral do processo de construção de sistemas de C-KBQA e como as principais abordagens são realizadas. Os trabalhos usam duas abordagens principais: Análise Semântica e Redes Neurais. No entanto, nos últimos anos, a combinação dessas duas abordagens tornou-se o estado da arte (chamada de análise semântica baseada em redes neurais). Por fim, foi discutido como a área C-KBQA está em crescimento e que novos sistemas de C-KBQA ou módulos que tentam melhorar partes das etapas de C-KBQA são prováveis de serem criados nos próximos anos.

Como trabalhos futuros estão a análise detalhada de conjuntos de dados para C-KBQA e as métricas de avaliação mais usadas nesse cenário, que ainda são desafios em aberto. Por fim, um panorama das publicações e dos locais mais relevantes através dos anos mostrará como a área do C-KBQA vem crescendo e recebendo mais atenção.

Agradecimentos

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

Referências

Agarwal, P., Ramanath, M., and Shroff, G. (2019). Retrieving relationships from a knowledge graph for question answering. In Azzopardi, L., Stein, B., Fuhr, N., Mayr, P.,

Hauff, C., and Hiemstra, D., editors, *Advances in Information Retrieval*, pages 35–50, Cham. Springer International Publishing.

Bao, J., Duan, N., Yan, Z., Zhou, M., and Zhao, T. (2016). Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514, Osaka, Japan. The COLING 2016 Organizing Committee.

Bhutani, N., Zheng, X., and Jagadish, H. V. (2019). Learning to answer complex questions over knowledge bases with query composition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 739–748, New York, NY, USA. Association for Computing Machinery.

Bhutani, N., Zheng, X., Qian, K., Li, Y., and Jagadish, H. (2020). Answering complex questions by combining information from curated and extracted knowledge bases. In *Proceedings of the First Workshop on Natural Language Interfaces*, pages 1–10, Online. Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bordes, A., Usunier, N., Chopra, S., and Weston, J. (2015). Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, abs/1506.02075.

Chen, X., Jia, S., and Xiang, Y. (2020). A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948.

Croft, W. B., Metzler, D., and Strohman, T. (2010). *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Publishing, USA.

Cui, W., Xiao, Y., Wang, H., Song, Y., Hwang, S., and Wang, W. (2019). KBQA: learning question answering over QA corpora and knowledge bases. *CoRR*, abs/1903.02419:565–576.

da Silva, J. W. F., Venceslau, A. D. P., Sales, J. E., Maia, J. G. R., Pinheiro, V. C. M., and Vidal, V. M. P. (2020). A short survey on end-to-end simple question answering systems. *Artificial Intelligence Review*, 53(7):5429–5453.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In McIlraith, S. A. and Weinberger, K. Q., editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818, New Orleans, Louisiana, USA. AAAI Press.

Dimitrakis, E., Sgontzos, K., and Tzitzikas, Y. (2019). A survey on question answering systems over linked data and documents. *Journal of Intelligent Information Systems*, 55:1–27.

Ding, J., Hu, W., Xu, Q., and Qu, Y. (2019). Leveraging frequent query substructures to generate formal queries for complex question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th*

International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2614–2622, Hong Kong, China. Association for Computational Linguistics.

Hao, Z., Wu, B., Wen, W., and Cai, R. (2019). A subgraph-representation-based method for answering complex questions over knowledge bases. *Neural Networks*, 119:57–65.

Höffner, K., Walter, S., Marx, E., Usbeck, R., Lehmann, J., and Ngonga Ngomo, A.-C. (2017). Survey on challenges of question answering in the semantic web. *Semantic Web*, 8(6):895–920.

Hu, S., Zou, L., and Zhang, X. (2018). A state-transition framework to answer complex questions over knowledge base. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2098–2108, Brussels, Belgium. Association for Computational Linguistics.

Hua, Y., Li, Y.-F., Haffari, G., Qi, G., and Wu, T. (2020a). Few-shot complex knowledge base question answering via meta reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5827–5837, Online. Association for Computational Linguistics.

Hua, Y., Li, Y.-F., Haffari, G., Qi, G., and Wu, W. (2020b). Retrieve, program, repeat: Complex knowledge base question answering via alternate meta-learning. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3679–3686, Virtual, Japan. International Joint Conferences on Artificial Intelligence Organization. Main track.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. (2020). A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, abs/2002.00388.

Jia, R. and Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Vancouver, Canada. Association for Computational Linguistics.

Kartsaklis, D., Pilehvar, M. T., and Collier, N. (2018). Mapping text to knowledge graph entities using multi-sense LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1959–1970, Brussels, Belgium. Association for Computational Linguistics.

Liu, Y., Wan, Y., He, L., Peng, H., and Yu, P. S. (2020). Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning.

Lukovnikov, D., Fischer, A., and Lehmann, J. (2019). Pretrained transformers for simple question answering over knowledge graphs. In *International Semantic Web Conference*, volume abs/2001.11985, pages 470–486. Springer.

Luo, K., Lin, F., Luo, X., and Zhu, K. (2018). Knowledge base question answering via encoding of complex query graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2185–2194, Brussels, Belgium. Association for Computational Linguistics.

McCoy, R. T., Min, J., and Linzen, T. (2019). Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance. *arXiv preprint arXiv:1911.02969*, abs/1911.02969:217–227.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8, New York, NY, USA. Association for Computing Machinery.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Miller, A., Fisch, A., Dodge, J., Karimi, A.-H., Bordes, A., and Weston, J. (2016). Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.

Namaki, M. H., Chowdhury, F. A., Islam, M., Doppa, J., and Wu, Y. (2017). Learning to speed up query planning in graph databases. *Proceedings of the International Conference on Automated Planning and Scheduling*, 27(1):9.

Neiva, F. W., David, J. M. N., Braga, R., and Campos, F. (2016). Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature. *Information and Software Technology*, 72:137–150.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18.

Petticrew, M. and Roberts, H. (2006). Systematic reviews in the social sciences: a practical guide. 2006. *Malden USA: Blackwell Publishing CrossRef Google Scholar*, 6:304–305.

Rodrigo, A. and Penas, A. (2017). A study about the future evaluation of question-answering systems. *Knowledge-Based Systems*, 137:83–93.

Saha, A., Pahuja, V., Khapra, M., Sankaranarayanan, K., and Chandar, S. (2018). Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):9.

Schumacher, E., Mayfield, J., and Dredze, M. (2020). Cross-lingual transfer in zero-shot cross-language entity linking.

Shin, S. and Lee, K.-H. (2020). Processing knowledge graph-based complex questions through question decomposition and recomposition. *Information Sciences*, 523:234–244.

Tong, P., Zhang, Q., and Yao, J. (2019). Leveraging domain context for question answering over knowledge graph. *Data Science and Engineering*, 4(4):323–335.

Trivedi, P., Maheshwari, G., Dubey, M., and Lehmann, J. (2017). Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218, Cham. Springer, Springer International Publishing.

Wu, L., Wu, P., and Zhang, X. (2019a). A seq2seq-based approach to question answering over knowledge bases. In *Joint International Semantic Technology Conference*, pages 170–181, Singapore. Springer, Springer Singapore.

Wu, P., Zhang, X., and Feng, Z. (2019b). A survey of question answering over knowledge base. In *China Conference on Knowledge Graph and Semantic Computing*, pages 86–97, Singapore. Springer, Springer Singapore.

Yang, Y. and Chang, M.-W. (2015). S-MART: Novel tree-based structured learning algorithms applied to tweet entity linking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 504–513, Beijing, China. Association for Computational Linguistics.

Yin, W., Ge, W., and Wang, H. (2014). Cdqa: An ontology-based question answering system for chinese delicacy. In *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*, pages 1–7, Shenzhen, China. IEEE.

Zhang, Y., Dai, H., Kozareva, Z., Smola, A., and Song, L. (2018). Variational reasoning for question answering with knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):6069–6076.

Zheng, C. and Kordjamshidi, P. (2020). SRLGRN: Semantic role labeling graph reasoning network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8881–8891, Online. Association for Computational Linguistics.

Part II

V Jornada de Descrição do Português

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Efeitos da variação linguística na decisão lexical

Victor Renê Andrade Souza[1], **Raquel Meister Ko. Freitag**[2]

[1]Programa de Pós-Graduação em Letras – Centro de Educação e Ciências Humanas
Universidade Federal de Sergipe (UFS) – São Cristóvão – SE – Brazil

[2]Departamento de Letras Vernáculas – Centro de Educação e Ciências Humanas
Universidade Federal de Sergipe (UFS) – São Cristóvão – SE – Brazil

victor.andrade573@gmail.com, rkofreitag@academico.ufs.br

Abstract. *This study presents a new version of the lexical decision test designed to capture the social appreciation of variable phonological phenomena [Freitag e Souza 2019]. The effects of lexical decision were tested in standard and non-standard variants of the phenomena of monophthongtion descending and increasing, denasalization of final unstressed nasal diphthong and palatalization of alveolar stops, in progressive and regressive environment, in a sample of 25 university students from Sergipe. The results reinforce what has already been identified in the previous study: stereotype and marker type variants are significantly associated with non-word; while the indicator variants do not show this relation.*

Resumo. *Este estudo apresenta uma nova versão do teste de decisão lexical desenvolvido para captar a apreciação social de fenômenos fonológicos variáveis [Freitag e Souza 2019]. Os efeitos de decisão lexical foram testados nas variantes padrão e não padrão dos fenômenos de monotongação de ditongo decrescente e crescente, desnasalização de ditongo nasal átono final e palatalização de oclusivas alveolares, em ambiente progressivo e regressivo, em uma amostra de 25 universitários de Sergipe. Os resultados reforçam o que já foi identificado no estudo anterior: as variantes do tipo estereótipo e marcador são significativamente associadas à não palavra; enquanto as variantes indicador não apresentam essa relação.*

1. Introdução

O campo do processamento da variação linguística tem demandado o desenvolvimento de novas técnicas de coleta de dados e métodos de análise, com o objetivo de identificar a relação entre a indexação social e um dado traço sociolinguístico variável. Nessa direção, o foco na consciência linguística, em suas diferentes dimensões, tem sido mobilizado no desenvolvimento de coletas de dados, como a consciência lexical. Categorizar em palavra ou não palavra itens como *parede* e *repede* é tarefa de fácil resposta entre falantes do português brasileiro. O paradigma de decisão lexical tem ampla aplicabilidade, sendo utilizado na investigação de processos como a natureza do léxico mental, efeitos de frequência de palavras, efeitos de vizinhança, medida de *priming* e efeitos de contexto, e como um índice de deficiências após dano cerebral [Goldinger 1996].

Ao introduzirmos neste processo o efeito de um traço fonológico variável, podemos perceber, de maneira indireta, o modo como os falantes de um dado grupo avaliam socialmente este traço. Fenômenos fonológicos variáveis na língua são avaliados de

modo diferente a depender do nível de consciência social atrelado ao traço linguístico. Tradicionalmente, a sociolinguística se pauta em uma classificação ternária da apreciação social de traços variáveis [Labov 2008]: i) indicadores, traços que estão abaixo do nível da consciência e por isso possuem pouca força avaliativa; ii) marcadores, sensíveis à estratificação social e estilística; e iii) estereótipos, traços linguísticos socialmente conscientes e marcados.

Um teste de decisão lexical voltado à identificação da avaliação social de traços fonológicos variáveis (Figura 1) mostrou que diferentes níveis de avaliação social atrelados a traços linguísticos interferem na decisão em palavra ou não palavra da língua [Freitag e Souza 2019].

Processo	Exemplo		Padrão na comunidade		Ocorrência		Tipo de avaliação
			- monitorado	+ monitorado	dialetal	social	
Monotongação -ow	 cenoura	cen/ow/ra	*nunca*	*frequente*	não	não	**Indicador**
		cen/o_/ra	*sempre*	*às vezes*			
Monotongação -aj, -ej	 caixa	c/aj/xa	*às vezes*	*frequente*	não	não	
		c/a_/xa	*frequente*	*às vezes*			
Palatalização regressiva	 vestido	ves/t/ido	*frequente*	*frequente*	sim	não	**marcador**
		ves/tʃ/ido	*às vezes*	*às vezes*			
Desnasalização ditongo final átono	 vagem	vag/eN/	*frequente*	*sempre*	não	sim	**estereótipo**
		vag/e_/	*às vezes*	*raro*			
Palatalização progressiva	oito	oi/t/o/	*frequente*	*sempre*	sim	sim	
		oi/tʃ/o	*às vezes*	*raro*			

Figura 1. Traços variáveis controlados em [Freitag e Souza 2019]

As palavras que continham os traços variáveis controlados em sua variante estigmatizada receberam mais julgamentos de não palavra do que palavras não afetadas pelos fenômenos. Os resultados do estudo experimental quanto à tarefa de discriminação e ao tempo de resposta apontam que as variantes do tipo estereótipo negativo e marcador são associadas à não palavra; enquanto as variantes do tipo indicador e esterótipo positivo são associados à palavra.

Estes resultados seguem a mesma direção de outros estudos. Variantes não canônicas no inglês americano são mais custosas em termos de processamento e são associadas em maior percentual à categoria de não palavra [Viebahn e Luce 2018, Viebahn e Luce 2020]. Em variedades do inglês americano, formas com redução fonética demandam maior esforço cognitivo [Tucker e Warner 2007, Tucker 2011], e juizes-ouvintes tendem a classificar como não palavra estímulos que contêm formas linguísticas de baixo prestígio social [Monteserín e Zevin 2016].

Assim, o aprimoramento de tarefas experimentais que se valem de teste de decisão lexical para desvelar o grau de apreciação social de um fenômeno fonológico variável, a partir do julgamento de palavra ou não palavra, pode ampliar o poder explanatório dos estudos de processamento sociolinguístico. Neste texto, apresentamos uma nova versão do teste de discriminação desenvolvido para captar a apreciação social de cinco fenômenos variáveis [Freitag e Souza 2019]; foi modificado o parâmetro de produção dos estímulos (uma locutora, em alinhamento a outros estudos, em vez de locutor, como no teste original), foram corrigidos contextos ambíguos e sem pareamento nos itens constantes do teste. Outra diferença em relação ao teste original foi a aplicação online, ao invés de coleta pre-

sencial em laboratório. A reprodução do experimento, em alinhamento ao movimento Ciência Aberta, contribui para o aprimoramento do teste e validação do paradigma de decisão lexical para o estudo do processamento da variação linguística. Mais um aspecto a ser considerado foi o fato de que este teste foi desenvolvido e aplicado em meio ao distanciamento físico imposto pela pandemia de covid-19, limitado aos recursos acessíveis (o que impactou diretamente na produção dos áudios e na seleção de uma plataforma para o experimento de acesso aberto).

2. Caracterização dos fenômenos-alvo

Foram testados os efeitos da variação linguística na tarefa de decisão lexical nos mesmos cinco fenômenos variáveis (com acréscimo da monotongação de ditongo crescente) na comunidade de fala de Aracaju, Sergipe, Brasil, com diferenças quanto ao nível de consciência social, já testados anteriormente (Figura 1).

O processo de monotongação consiste no apagamento da semivogal do ditongo. No português brasileiro, podem ser monotongados ditongos decrescentes, vogal seguida de glide, e ditongos crescentes, compostos por glide seguido de vogal. A monotongação de ditongo decrescente pode ocorrer com o apagamento do glide palatal [j], como em [kaʃɔ], ou do velar [w], como em [senorɔ]. O fenômeno é estável em todas as regiões do Brasil, sem sensibilidade social ou dialetal [Araujo e Borges 2018]. O apagamento do glide velar é visto como uma mudança consolidada no português brasileiro, sem restrições linguísticas ou sociais, apresentando percentuais elevados em todos os contextos linguísticos [Cristofolini 2011, Freitas 2017, Silveira 2019], inclusive em situações de maior monitoramento estilístico como a leitura em voz alta [Hora e Aquino 2012, Machado 2018]. A monotongação de /aj/ e de /ej/ apresenta restrições estruturais relativas ao contexto fonológico seguinte, isto é, o apagamento da semivogal depende do som que vem depois do ditongo. A redução de /ej/ se comporta como um fenômeno tipicamente variável, com frequências distribuídas entre ditongo preservado e vogal simples. O processo tem motivação estrutural relacionada principalmente ao contexto fonológico seguinte constituído por tepe [ɾ] e, com menor força, por consoantes palatais [ʃ, ʒ] [Haupt 2011, Toledo 2011, Freitas 2017, Silveira 2019]. A monotongação de /aj/ possui dois contextos propícios específicos: em sílaba aberta, em contexto seguido de consoante palato-alveolar [ʃ], sendo o ditongo preservado nos demais contextos; e em sílaba fechada, quando a fricativa final é palatalizada, ocorrendo principalmente em falantes da região Sul do Brasil [Haupt 2011, Silveira 2019].

No caso dos ditongos crescentes, estudos apontam para i) variação livre entre ditongo, monotongo e hiato [Silva e Faria 2014] e ii) redução da semivogal relacionada à saliência articulatória entre a vogal e a semivogal do ditongo [Hora e Aquino 2012]. A monotongação de ditongos crescentes constituídos por vogal e semivogal salientes em termos articulatórios, como [polisɔ], são estereótipos negativos associados a falantes de baixa escolaridade e da zona rural. A redução da semivogal em ditongos crescentes constituídos por vogal e semivogal mais próximas do ponto de vista articulatório, como [sɛɾɪ], não é sensível à avaliação social [Araujo e Borges 2018].

O processo de desnasalização de ditongo nasal átono final decorre do apagamento do segmento nasal em nomes, como [masaʒɪ], e em verbos de terceira pessoa, como [falaɾʊ]. Em nomes, o apagamento do segmento nasal é associado a aspectos sociais, como

escolaridade e ruralidade, com sensibilidade ao contexto de monitoramento estilístico; a variante desnasalizada é considerada, conforme a tipologia de apreciação social, um marcador [Gomes et al. 2013, Gomes 2017]. Nos verbos, também com relativa sensibilidade ao contexto de monitoramento estilístico, a variante desnasalizada tem ocorrência estável em todo o português brasileiro, com interferência nas relações morfossintáticas.

A palatalização das oclusivas alveolares /t/ e /d/ no português brasileiro pode ocorrer em dois ambientes fonológicos, anterior e posterior à vogal [i], que apresentam comportamentos sociolinguísticos e níveis de apreciação social distintos. A palatalização regressiva é desencadeada por vogal alta [i] posterior às consoantes /t, d/, que resultam nas realizações palatais, como [dʃikɔ], [tʃipʊ]. A variante palatalizada nesse ambiente é reconhecida como de prestígio, caracterizando-se como estereótipo positivo; a variante oclusiva, por sua vez, é associada ao dialeto nordestino e rural. Em Sergipe, onde a variante oclusiva é predominante, percebe-se uma mudança incipiente em direção à variante palatalizada [Souza Neto 2008, Freitag e Santos 2016, Ribeiro et al. 2018].

A palatalização progressiva ocorre quando desencadeada por glide palatal [j] anterior às consoantes /t, d/, resultando nas realizações palatais, como [pejtʃʊ]. Essa realização é associada socialmente a pessoas mais velhas, com baixa escolaridade e da zona rural, o que, segundo a tipologia da apreciação social, a aponta como estereótipo negativo. Em Sergipe, quanto à produção sociolinguística, estudos [Souza Neto 2008, Freitag e Santos 2016, Ribeiro et al. 2018] apontam a redução dessa variante em face à implementação da variante oclusiva, que se caracteriza como indicador.

3. Método

3.1. Instrumento

A tarefa experimental desenvolvida constituiu-se no julgamento de estímulos linguísticos em "palavra" ou "não palavra" do português. Para compor o conjunto de estímulos, as palavras foram definidas segundo critérios de familiaridade (produtividade na língua), de tamanho (mesmo número de sílabas e tamanho) e de barramento de palavras que pudessem apresentar outros fenômenos variáveis. Estes mesmos critérios foram utilizados no estudo anterior [Freitag e Souza 2019]. Aprimoramos principalmente a padronização entre os pares, de modo que o contexto alvo estivesse na mesma posição.

Os áudios foram gravados por meio de aparelho celular *Galaxy J5 Prime*, em formato de som *.wav* e taxa de compressão de 1411kbps. O conjunto de palavras foi enunciado por uma única locutora, reconhecida como representativa da comunidade de fala de Aracaju, Sergipe, segundo seus pares, na seguinte frase-veículo: "Eu falo —— devagar", a fim de favorecer a realização não-final. Após esta etapa, os áudios foram recortados no software Audacity para produção dos estímulos para a tarefa.

Do mesmo modo que no estudo anterior [Freitag e Souza 2019], os estímulos foram divididos em três conjuntos (Figura 2):

- estímulos-alvos: composto por cinco fenômenos variáveis na comunidade de fala de Aracaju, Sergipe, Brasil (variante padrão e não padrão de monotongação de ditongo decrescente e crescente, desnasalização de ditongo nasal átono final e palatalização de oclusivas alveolares em ambiente progressivo e regressivo);

- distratores: constituído por palavras do português que barram a variação linguística;
- pseudopalavras: composto por palavras com a fonotaxe do português brasileiro, que, no entanto, não se configuram lexicalmente como palavra.

Monotongação						**Desnasalização de ditongo nasal átono final**	
Ditongo crescente		**Ditongo decrescente**					
		/ow/		**/ej/ e /aj/**			
Ditongo	**Monotongo**	**Ditongo**	**Monotongo**	**Ditongo**	**Monotongo**	**Nasal**	**Desnasalizada**
delíc/ja/	políç/_a/	cen/ow/ra	vass/o_/ra	p/ej/xe	f/e_/xe	corag/ʒeN/	garag/e_/
paciênc/ja/	ciênç/_a/	/ow/ro	c/o_/ro	/ej/xo	qu/e_/xo	passag/ʒeN/	mensag/e_/
comérc/jo/	negóç/_o/	l/ow/co	p/o_/co	c/aj/xa	f/a_/xa	imag/ʒeN/	viag/e_/
cár/je/	sér/_e/	p/ow/pa	r/o_/pa	qu/ej/jo	b/e_/jo	ont/ʒeN/	hom/e_/
superfíc/je/	espéc/_e/	l/ow/ça	tr/o_/xa	dinh/ej/ro	brasil/e_/ro	ord/ʒeN/	jov/e_/
árd/wo/	vác/_o/	p/ow/so	rep/o_/so	gelad/ej/ra	band/e_/ra	órf/ãw/	órg/_u/
Palatalização				**Distratores**		**Pseudopalavras**	
Ambiente regressivo		**Ambiente progressivo**					
Oclusivo	**Palatal**	**Oclusiva**	**Palatal**				
ves/t/ido	des/tʃ/ino	oi/t/o	dezoi/ʃt/o	mola	tabela	tapi	pimada
tar/d/e	par/tʃ/e	pei/t/o	respei/tʃ/o	sapo	palito	sifo	navela
ban/d/ido	men/dʒ/igo	prefei/t/ura	lei/tʃ/ura	bola	favela	tixo	dibata
gen/t/e	men/tʃ/e	mui/t/a	mui/tʃ /o	rato	cinema	pila	butove
/t/ipo	/dʒ/ica	biscoi/t/o	doi/dʒ/o	gelo	cabelo	lobe	boreza
meta/d/e	sauda/dʒ/e	coi/t/ado	cui/dʒ/ado	cubo	barata	cafa	decato

Figura 2. Conjunto de estímulos linguísticos do teste de decisão lexical.

O teste foi estruturado no software OpenSesame [Mathôt et al. 2012] (Figura 3). Testamos a compatibilidade do teste na versão online através da ferramenta OSWeb, que permite verificar se o experimento é compatível para a versão online e executar o teste localmente no navegador.

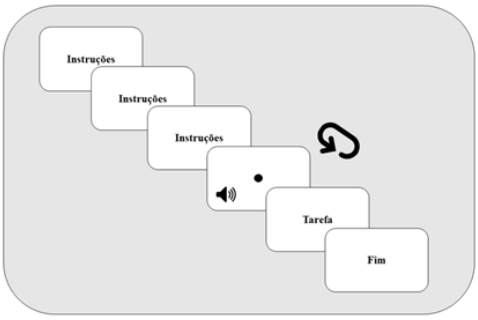


Figura 3. Estrutura do teste de decisão lexical.

3.2. Procedimentos de coleta

O experimento foi aplicado através da plataforma de gerenciamento de experimentos Just Another Tool for Online Studies – JATOS [Lange et al. 2015]. Essa plataforma, open-source, gratuita e com interface gráfica do usuário, permite configuar e executar estudos online, possibilitando total controle sobre o acesso aos dados dos resultados, garantindo que cada participante só realize o teste uma vez.

O experimento foi exportado do OpenSesame e importado no JATOS. Além do arquivo do OpenSesame, foi importado também formulário para o termo de consentimento livre e esclarecido e para a coleta de informações sociais do informante, como sexo, cidade de realização do teste.

O teste foi transformado em link através do gerenciador da plataforma e compartilhado com os participantes da pesquisa através do Ambiente Virtual de Aprendizagem (AVA) da Universidade Federal de Seripe (UFS). Junto ao link, os participantes foram instruídos de que o teste só poderia ser realizado uma vez, exclusivamente em computador ou notebook e utilizando fones de ouvido. Além disso, os participantes foram alertados sobre evitar qualquer tipo de distração (rede social, televisão, etc.) ao longo da realização do teste.

3.3. Participantes

Participaram da pesquisa 25 estudantes de graduação do curso de Letras do Centro de Educação Superior a Distância (CESAD) da Universidade Federal de Sergipe (UFS) – de vários municípios do estado – que estavam cursando a disciplina Fonologia da Língua Portuguesa (LETRV0057), componente curricular do primeiro período do curso. Restringimos a distribuição do link apenas aos alunos da disciplina devido limitações do servidor. O disparo do instrumento de coleta foi antes do início das aulas, de modo que os participantes não tinham conhecimento teórico sobre os fenômenos variáveis.

3.4. Tratamento estatístico dos dados

Após a coleta, os dados foram tratados quantitativamente. Foi calculada a proporção de decisão entre palavra e não palavra para cada conjunto de variantes dos processos sob análise, e teste-t entre as médias de tempo de resposta e as variantes dos processos. A visualização gráfica foi desenvolvida na plataforma R [Wickham et al. 2019, Wickham 2016, Kassambara 2020a, Kassambara 2020b].

4. Resultados e discussões

Os resultados de decisão lexical aos estímulos seguem a tendência identificada no estudo anterior de que variantes avaliadas negativamente tendem a ser consideradas como “não palavras” da língua, enquanto estereótipos positivos ou com distribuição não saliente não apresentam diferenças em relação às decisões lexicais (Figura 4).

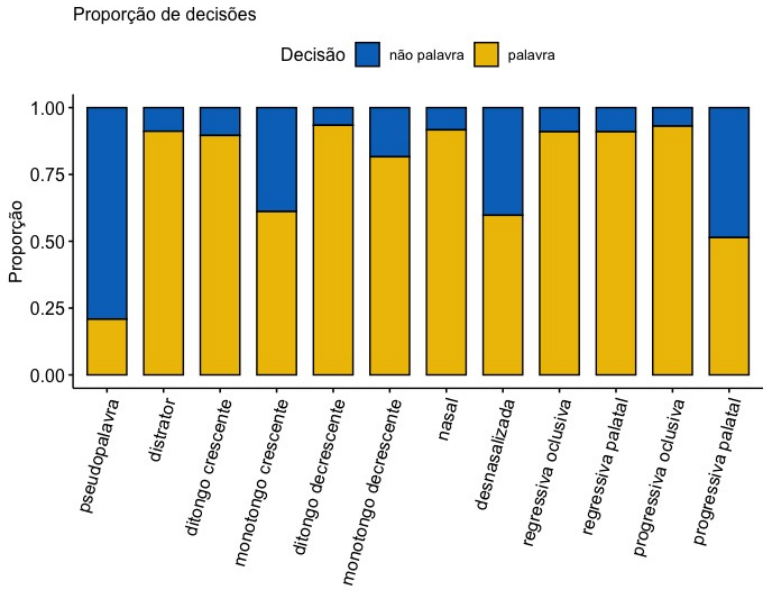


Figura 4. Proporção de decisão lexical para cada tipo de estímulo.

O percentual de decisão lexical positiva para os estímulos distratores é de 91%; para as pseudopalavras o percentual é de 21%. Em relação à comparação entre variante padrão e não padrão de cada fenômeno, os percentuais de julgamento como não palavra são maiores para os itens com a variante não padrão com avaliação social negativa. A forma monotongada de ditongo crescente apresentou um percentual de 39% de decisões lexicais negativas; o ditongo crescente preservado, por sua vez, foi avaliado como não palavra em apenas 10% dos casos. A variante desnasalizada do ditongo nasal átono final apresentou percentual de julgamento negativo maior do que o da variante nasal, com percentuais de 40% e 8%, respectivamente. Na mesma direção, a realização palatal em ambiente progressivo de /t/ e /d/ apresentou maior percentual de decisões como não palavra (49%) se comparada à variante oclusiva (7%). A tendência observada é a de que as variantes resultantes de fenômenos variáveis estigmatizados socialmente (monotongação de ditongo crescente, desnasalização de ditongo nasal átono final e palatalização progressiva) são associadas à não palavra, enquanto variantes sem sensibilidade à avaliação social não apresentam essa relação (monotongação de ditongo decrescente).

O esforço cognitivo do processamento de uma variante não esperada para o contexto ou estigmatizada pode ser medido quanto ao tempo de resposta à tarefa de decisão lexical, uma medida on-line do processamento também controlada no estudo anterior (Figura 5). Itens com variantes estigmatizadas socialmente demandam maior tempo de processamento, medido em termos de maior tempo de resposta, do que itens com variantes sem sensibilidade à avaliação social.

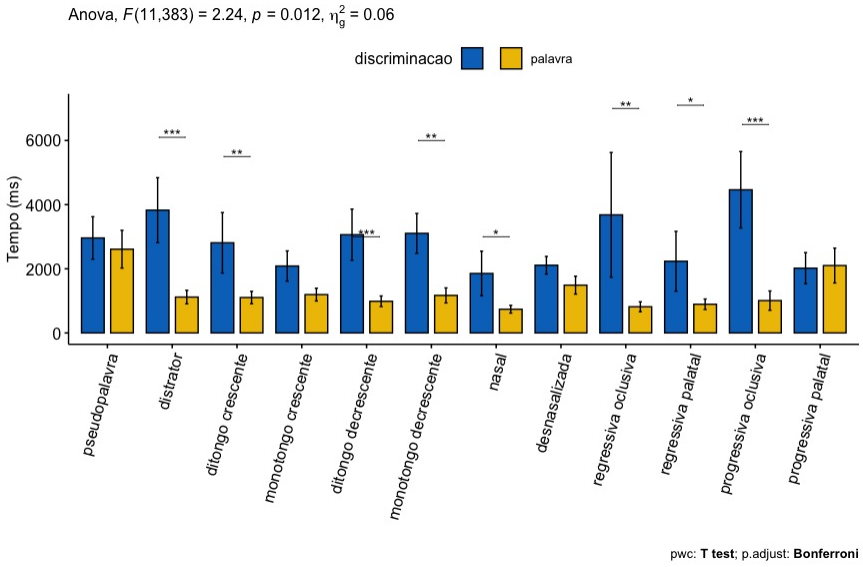


Figura 5. Tempo médio de resposta para cada tipo de estímulo.

Os resultados apontam que os itens com as variantes não padrão demandaram maior tempo de processamento, se comparados aos itens com as variantes padrão. Todas as variantes tidas como não padrão apresentam tempo de resposta superior à variante tida como padrão em situações formais na comunidade: a realização de ditongo decrescente tem tempo médio de 556ms, e a variante monotongada, 549.50ms; o ditongo crescente tem tempo médio de 553.50ms, e a variante monotongada tem 748ms; a realização da átona final nasal tem tempo médio de 427ms, e a variante desnasalizada 864.50ms; a realização de /t, d/ em ambiente seguinte à vogal alta, regressiva oclusiva, apresenta tempo médio de 478ms; e a realização palatal, 481.50ms; a variante oclusiva em ambiente progressivo tem tempo médio de 561.50ms, a palatal tem média de 786.50ms. Esse resultado reforça a hipótese de que o processamento da variação linguística, quando em variantes alvo de avaliação social, são mais custosas em tempo de processamento [Viebahn e Luce 2018, Viebahn e Luce 2020].

5. Considerações finais

Os resultados do teste de decisão lexical com estímulos em áudio envolvendo traços fonológicos variáveis com diferentes padrões de avaliação social reforçam o que já foi identificado no teste anterior, evidenciando que variantes não padrão demandam maior esforço cognitivo de processamento e tendem a ser categorizadas como não palavra.

Este estudo contribui para a replicabilidade (entendida aqui como a replicação de um estudo com a mesma abordagem analítica) como também para a generalização (entendida aqui como a mesma abordagem analítica em conjuntos de dados diferentes), em alinhamento aos preceitos de Ciência Aberta [Freitag et al. 2021], ao mesmo tempo que contribui para o desenvolvimento de técnicas que possam automatizar o processamento da linguagem, com a identificação de traços linguísticos variáveis sensíveis à avaliação social.

Referências

Araujo, A. S. e Borges, D. K. V. (2018). Atitudes linguísticas de estudantes universitários: o fenômeno da monotongação em foco. *Tabuleiro de Letras*, 12:97–113.

Cristofolini, C. (2011). Estudo da monotongação de [ow] no falar florianopolitano: perspectiva acústica e sociolinguística. *Revista da ABRALIN*, 10(1):205–229.

Freitag, R., Tejada, J., Pinheiro, B., e Cardoso, P. (2021). Função na língua, generalização e reprodutibilidade. *Revista da ABRALIN*, pages 1–27.

Freitag, R. M. K. e Santos, A. d. O. (2016). Percepção e atitudes linguísticas em relação às africadas pós-alveolares em sergipe. *A Fala Nordestina: entre a sociolinguística e a dialetologia. São Paulo: Blucher*, pages 109–122.

Freitag, R. M. K. e Souza, V. R. A. (2019). Discriminação de palavras e efeitos da variação linguística. In *XII Symposium in Information and Human Language Technology and Collocates Events. Proceedings*.

Freitas, B. F. C. d. (2017). Estudo da monotongação de ditongos orais decrescentes na fala uberabense.

Goldinger, S. D. (1996). Auditory lexical decision. *Language and Cognitive Processes*, 11(6):559–568.

Gomes, C. A. (2017). Para além das ondas: um ponto de partida sobre o significado social da variação entre ditongo nasal átono final e vogal oral no português brasileiro. *diacrítica*, 31(1):20–20.

Gomes, C. A., Mesquita, C., e Fagundes, T. d. S. (2013). Revisitando a variação entre ditongos nasais finais átonos e vogais orais na comunidade de fala do rio de janeiro. *diacrítica*, 27(1):153–173.

Haupt, C. (2011). O fenômeno da monotongação nos ditongos [ai, ei, oi, ui] na fala dos florianopolitanos: uma abordagem a partir da fonologia de uso e da teoria dos exemplares.

Hora, D. d. e Aquino, M. d. F. S. (2012). Da fala para a leitura: análise variacionista. *Alfa: Revista de Linguística (São José do Rio Preto)*, 56:1099–1115.

Kassambara, A. (2020a). *ggpubr: 'ggplot2' Based Publication Ready Plots*. R package version 0.4.0.

Kassambara, A. (2020b). *rstatix: Pipe-Friendly Framework for Basic Statistical Tests*. R package version 0.6.0.

Labov, W. (2008). Padrões sociolinguísticos: Trad. *Marcos Bagno, Maria Marta Pereira Scherre e Caroline Rodrigues Cardos. São Paulo: Parábola Editorial,[1972]*.

Lange, K., Kühn, S., e Filevich, E. (2015). "just another tool for online studies"(jatos): An easy solution for setup and management of web servers supporting online studies. *PloS one*, 10(6):e0130834.

Machado, A. P. G. (2018). Variação linguística e leitura: fenômenos variáveis da fala na leitura em voz alta. *A Cor das letras*, 19(4Especial):196–218.

Mathôt, S., Schreij, D., e Theeuwes, J. (2012). Opensesame: An open-source, graphical experiment builder for the social sciences. *Behavior research methods*, 44(2):314–324.

Monteserín, M. L. e Zevin, J. D. (2016). Investigating the impact of dialect prestige on lexical decision. In *INTERSPEECH*, pages 2214–2218.

Ribeiro, C. C., São Cristóvão de Santana, S., e de Andrade Corrêa, B. T. R. (2018). Avaliação social da palatalização de/t, d/em sergipe. *A Cor das Letras*, 19(4Especial):109–123.

Silva, T. C. e Faria, I. (2014). Percursos de ditongos crescentes no português brasileiro. *Letras de Hoje*, 49(1):19–27.

Silveira, L. M. d. (2019). Monotongação em uso no português do sul do brasil.

Souza Neto, A. F. d. (2008). Realizações dos fonemas/t/e/d/em aracaju sergipe.

Toledo, E. E. (2011). A monotongação do ditongo decrescente/ej/em amostra de recontato de porto alegre.

Tucker, B. V. (2011). The effect of reduction on the processing of flaps and/g/in isolated words. *Journal of Phonetics*, 39(3):312–318.

Tucker, B. V. e Warner, N. (2007). Inhibition of processing due to reduction of the american english flap. In *Proceedings of the 16th international congress of phonetic sciences*, pages 1949–1952.

Viebahn, M. C. e Luce, P. A. (2018). Increased exposure and phonetic context help listeners recognize allophonic variants. *Attention, Perception, & Psychophysics*, 80(6):1539–1558.

Viebahn, M. C. e Luce, P. A. (2020). Where is the disadvantage for reduced pronunciation variants in spoken-word recognition? on the neglected role of the decision stage in the processing of word-form variation. *Language, Cognition and Neuroscience*, 35(3):339–359.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., e Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Palatalização na fala e na leitura de universitários sergipanos

Lucas Santos Silva[1], **Raquel Meister Ko. Freitag**[2]

[1]Programa de Pós-Graduação em Letras – Centro de Educação e Ciências Humanas
Universidade Federal de Sergipe (UFS) – São Cristóvão – SE – Brazil

[2]Departamento de Letras Vernáculas – Centro de Educação e Ciências Humanas
Universidade Federal de Sergipe (UFS) – São Cristóvão – SE – Brazil

lucas_riachao@hotmail.com, rkofreitag@academico.ufs.br

Abstract. *A comparative analysis of the palatalization of /t/ and /d/ in two independent and different-sized samples, composed of speech (n = 8.850) and read-aloud (n = 831) data from students at the Universidade Federal de Sergipe, is presented. Social variables displacement and time in the course, and linguistic variables anterior context, posterior context, tonicity, and sonority, were controlled in order to identifying whether the constraining effects of palatalization in speech also act on reading aloud data. The results point out that, even in distinct samples, the behavior of social and linguistic constrains shows stability in the change and propagation of palatalization process in the community.*

Resumo. *Apresentamos uma análise comparativa acerca da palatalização de /t/ e /d/ em duas amostras independentes, e com tamanho distinto, compostas por dados de fala (n = 8.850) e de leitura em voz alta (n = 831) de estudantes da Universidade Federal de Sergipe. Controlamos as variáveis sociais deslocamento e tempo no curso, e as variáveis linguísticas contexto anterior, contexto posterior, tonicidade e sonoridade, com o objetivo de identificar se os efeitos condicionantes da palatalização da fala atuam na leitura em voz alta. Os resultados sinalizam que, mesmo em amostras distintas, o comportamento dos fatores sociais e linguísticos apresentam estabilidade na mudança e propagação da palatalização na comunidade.*

1. Introdução

A palatalização regressiva de /t/ e /d/, como em tia[tʃia] ou dia[dʒia], é um traço fonético-fonológico que, no estado de Sergipe, passa por mudança incipiente com o aumento das variantes palatalizadas. O fenômeno foi descrito tanto em amostra de fala espontânea [Souza Neto 2008, Souza 2016, Freitag and Souza 2016, Corrêa 2019, Freitag et al. 2019], quanto em leitura em voz alta, com estudantes da educação básica [Pinheiro et al. 2017], assim como em estudos de percepção [Freitag and Santos 2016, Freitag 2020]. Os mesmos estudos apontam para o efeito dialetal condicionador da variação, com diferença na frequência entre moradores da capital sergipana, sendo o processo da palatalização mais frequente na capital [Souza 2016, Corrêa 2019].

Processos fonológicos da fala podem ser transpostos para a leitura em voz alta [Freitag and Sá 2019, Souza et al. 2020, Pinheiro et al. 2017] A identificação do fenômeno da palatalização na leitura em voz alta sinaliza que a variável emergente – a realização palatal – não é estigmatizada na comunidade. No escopo do projeto *Como*

fala, lê e escreve o universitário?, a fim de identificar se condicionadores internos e externos da palatalização em contextos da fala atuam na leitura em voz alta, controlamos as variáveis sociais deslocamento e tempo de curso, e as variáveis linguísticas contexto anterior, contexto posterior, tonicidade e sonoridade em dois conjuntos de dados: uma amostra composta pela leitura em voz alta por 36 estudantes universitários sergipanos [Silva 2021], e o conjunto de dados de palatalização de uma amostra composta por entrevistas sociolinguísticas com 64 estudantes universitários sergipanos [Corrêa 2019].

A comparação dos condicionadores pode contribuir para identificar padrões na fala e na leitura em voz alta para a emergência de fenômenos variáveis no português brasileiro, ampliando o escopo de técnicas de coleta de dados.

2. Método

O conjunto dos dados de palatalização na fala espontânea é composto por 8.550 contextos de /t/ e /d/ diante de /i/ [Corrêa 2019]. Destas observações, 7.543 (88,2%) referem-se à realização oclusiva. O conjunto dos dados de leitura em voz alta é composto por 831 contextos de /t/ e /d/ diante de /i/, dos quais 424 (51,02%) referem-se à realização oclusiva [Silva 2021]. Ambos os conjuntos de dados provêm de uma mesma comunidade de fala, a de estudantes universitários da Universidade Federal de Sergipe, estratificados quanto ao tempo do curso, ao sexo e ao tipo de deslocamento que realizam: D1, D2, D3.

- Deslocamento 1 é composto por estudantes que residem na grande Aracaju;
- Deslocamento 2 é composto por estudantes residentes do interior do estado de Sergipe (nascidos e criados) e que fazem o movimento pendular diário para a UFS;
- Deslocamento 3 é composto por estudantes nascidos e criados no interior do estado de Sergipe, mas que moram na grande Aracaju por causa de UFS.

As amostras são independentes, e com tamanho distinto. Para poder realizar a comparação dos efeitos condicionantes, foi calculada a taxa de realização da variável dependente por falante, na fala espotânea e na leitura em voz alta, considerando o estudande como unidade de análise. Utilizamos o teste de Wilcoxon para amostras não pareadas. Primeiro, consideramos a frequência em cada variável, e depois, em cada variável, frequência e estilo, com ajuste de Bonferroni para ambos os casos. Os gráficos foram produzidos no R [Wickham et al. 2019, Wickham 2016, Kassambara 2020a, Kassambara 2020b].

3. Resultados

3.1. Condicionamentos sociais

Considerando os deslocamentos dos universitários sergipanos, não há diferença estatisticamente significativa entre a taxa média de palatalização na fala e na leitura em voz alta (Figura 1).

Os estudantes que têm maior contato com a região da grande a Aracaju (D1 e D2) tendem a fazer maior uso das variantes palatalizadas tanto na fala quanto na leitura em voz alta. A comparação em cada um dos níveis de deslocamos entre a taxa de palatalização na fala e na leitura em voz alta é estatisticamente significativa. Os universitários oriundos do

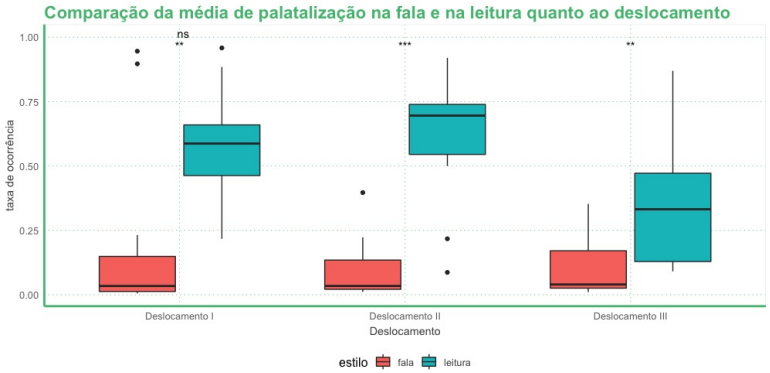


Figura 1. Comparação entre a taxa de realização de palatalização na fala e na leitura quanto ao deslocamento, com teste de Wilcoxon para o estilo, e comparação pareada entre os níveis

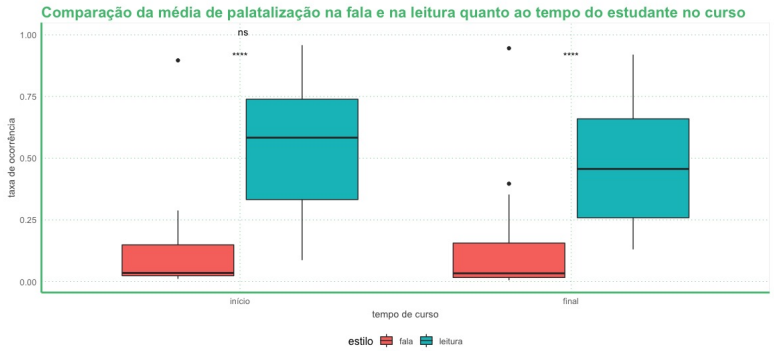


Figura 2. Comparação entre a taxa de realização de palatalização na fala e na leitura quanto ao tempo de curso, com teste de Wilcoxon para o estilo, e comparação pareada entre os níveis

interior do estado de Sergipe que foram morar na região da grande Aracaju apresentam a menor média na taxa de palatalização.

O tempo de curso pressupõe que o contato com a norma da comunidade pode afetar o padrão de uso da palatalização: universitários ao final do curso tendem a ter um comportamento linguístico diferente daqueles do início do curso. A comparação entre as taxas de palatalização na fala e na leitura em voz alta considerando o tempo de curso do estudante não se mostrou estatisticamente significativa (Figura 2). O tempo do curso não interfere na palatalização na leitura em voz alta. Mesmo não havendo diferença estatisticamente significativa, pelos valores das médias da taxa de ocorrência da palatalização, podemos inferir que os universitários do início do curso são os que fazem o maior uso das realizações palatalizadas. Novamente, há diferença estatisticamente significativa na comparação entre a fala e a leitura em cada um dos níveis da variável.

3.2. Condicionamentos linguísticos

O contexto linguístico antecedente ao ambiente apresenta diferença estatisticamente significativa na comparação entre a fala e a leitura em voz alta (Figura 3). Nos contextos de fricativas alveolares, a exemplo de *nos dizia* [nuzdʒiˈziə], *se tivéssemos* [s tʃiˈvɛsimʊs], as médias na taxa de palatalização na fala e na leitura em voz alta foram, respectivamente, 0,67 e 0,75, sem diferença estatisticamente significativa. Nos demais contextos, houve diferença estatisticamente significativa entre as médias de palatalização na leitura, todas superiores às encontradas na fala: fricativas alveolopalatais (*tivéssemos de* [tiˈvɛsimʊʒˈdʒɪ]) e glotais (*morte* [ˈmɔhtʃɪ]); vogais anteriores (*que tinham* [kɪ ˈtʃiɲẽʊ]) e vogais centrais (*atiraríamos* [a.tʃɪraˈɾiãmʊs]).

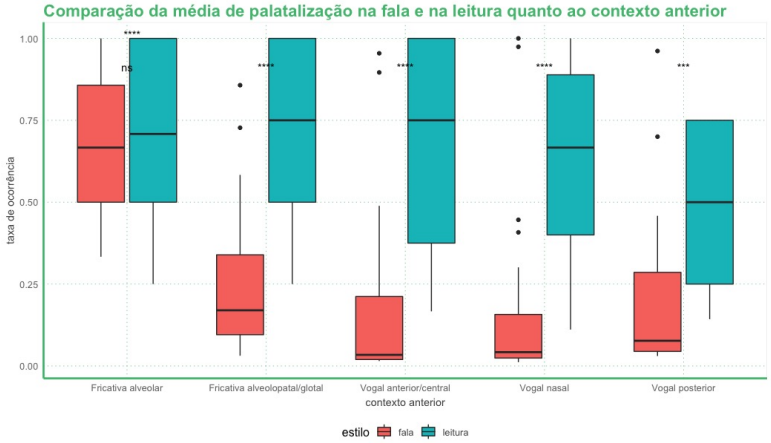


Figura 3. Comparação entre a taxa de realização de palatalização na fala e na leitura quanto ao contexto linguístico anterior, com teste de Wilcoxon para o estilo, e comparação pareada entre os níveis

Mesmo padrão de comportamento identificado no contexto linguístico sucedente ao ambiente, que também apresenta diferença estatisicamente significativa na comparação entre fala e leitura em voz alta (Figura 4). A diferença entre as médias de palatalização entre fala e leitura é mais acentuada nos ambientes seguidos por pausa (*balde* [ˈbawdʒɪ]) e por vogal (*apaixonamentente e...* [apaɪʃõnadaˈmẽtʃɪ e]), do que nos ambientes seguidos por consoante líquida ou fricativa (*atiraríamos* [atʃɪraˈɾiãmʊs] e *utilizando* [utʃiliˈzãdʊ]) e consoante oclusiva/nasal (*tipo* [ˈtɪpʊ] e *continham* [kõtʃiɲuãʊ]). O resultado segue o *continuum* para a mudança e propagação da palatalização, na fala e na leitura em voz alta, primeiro sendo as vogais e pausa em contexto seguinte as favorecedoras para a emergência do fenômeno em Sergipe [Corrêa 2019].

Quanto à tonicidade e à sonoridade, não há diferença estatisticamente significativa na taxa de palatalização na fala e na leitura. Na comparação entre os níveis de tonicidade (Figura 5), as médias na taxa de palatalização na leitura são superiores às da fala no postônico final (*morte*, *apaixonadamente*, *balde*, *pode*), pretônico (*continuam*, *antigamente*, *tivéssemos*, *dizia*) e tônico (*iludindo*, *ditas*, *tinham*). As médias na taxa de palatalização na fala e na leitura em voz alta são maiores no contexto tônico, corroborando a hipótese de que a emergência da palatalização se dá em contexto de sílaba tônica

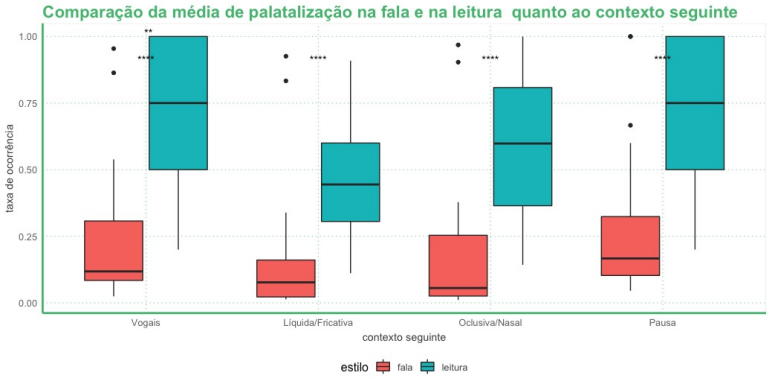


Figura 4. Comparação entre a taxa de realização de palatalização na fala e na leitura quanto ao contexto linguístico posterior, com teste de Wilcoxon para o estilo, e comparação pareada entre os níveis

[Câmara 1970].

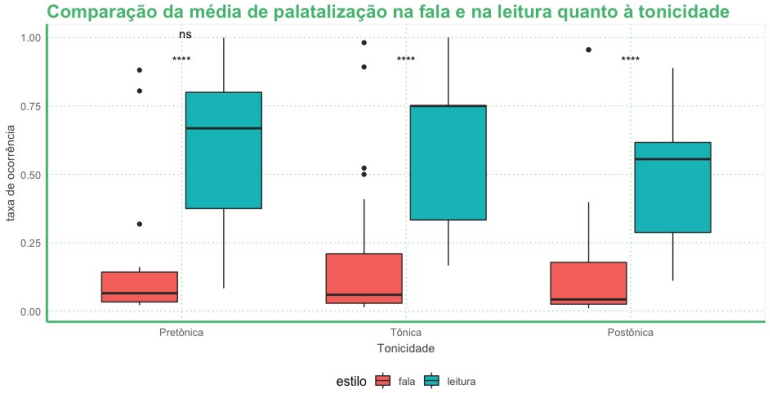


Figura 5. Comparação entre a taxa de realização de palatalização na fala e na leitura quanto à tonicidade da sílaba, com teste de Wilcoxon para o estilo, e comparação pareada entre os níveis

A diferença entre as amostras de fala e de leitura não é estatisticamente significativa quanto à sonoridade da consoante (Figura 6). As médias na taxa de palatalização apresentam semelhança em ambas as amostras: na fala, média = 0,12 para sonoras e média = 0,16 para surdas; para a leitura, média = 0,48 das sonoras e média = 0,59 das surdas. Em ambas as amostras, a consoante surda /t/ apresenta maior média de palatalização.

4. Discussão

Os resultados apresentados apontam que, embora haja diferença nos valores de frequência, com médias maiores na leitura do que na fala, o comportamento dos fatores sociais e linguísticos controlados apresentam estabilidade. A realização palatal é uma variante que tem aumentado sistematicamente (não em progressão exponencial), sendo

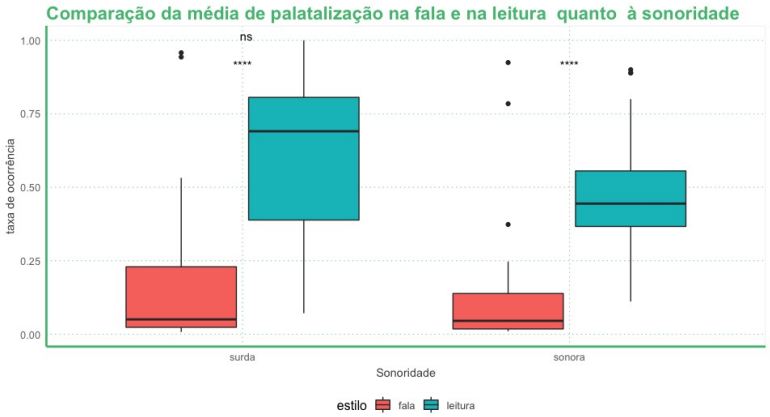


Figura 6. Comparação entre a taxa de realização de palatalização na fala e na leitura quanto à sonoridade da consoante, com teste de Wilcoxon para o estilo, e comparação pareada entre os níveis

socialmente prestigiada, conscientemente bem avaliada pela comunidade e associada aos falantes com alta escolaridade. Por isso, as maiores médias de frequência da palatalização na leitura em voz alta de universitários podem apresentar associação com o prestígio da variante e com o fato da tarefa de leitura ser altamente monitorada e consciente pelos estudantes.

Nas variáveis linguísticas, mesmo tendo diferenças estatisticamente significativas nos contextos seguinte e antecedente, os resultados são inconclusivos e não nos permitem fazer generalizações, uma vez que metodologias diferentes quanto aos dados de fala e de leitura foram adotadas. Para realizar a comparação, foi necessário realizar amalgamações entre níveis. Além disso, há menos itens lexicais nos dados de leitura ou itens sobrepostos: por exemplo, na leitura, a fricativa alveolar ocorre em seis contextos, dos quais cinco eram de consoante sonora, o que pode ter levado a uma diferença e não permite estabelecer uma generalização acerca da comparação entre os contextos. Tanto na fala quanto na leitura, o contexto das fricativas alveolares [s, z] é o que mais favorece a ocorrência de palatalização, resultado que segue a direção de outros estudos já desenvolvidos em Sergipe [Souza Neto 2008, Souza 2016, Corrêa 2019].

Ao receber alunos de diversas regiões geográficas de Sergipe, a Universidade Federal de Sergipe configura-se como um ambiente de troca de experiência. A inserção de universitários de outras comunidades do interior de Sergipe à grande Aracaju e, sobretudo, à Universidade Federal de Sergipe, contribui para o aumento da taxa de palatalização, uma vez que os estudantes assumem novas experiências linguísticas enquanto falantes.

No contexto altamente monitorado, como é o da leitura em voz alta, a média na taxa de realização de palatalização é maior em estudantes residentes no interior do estado de Sergipe, o que vai de encontro aos estudos já desenvolvidos acerca da palatalização em Sergipe com dados de fala e que apontam que o fenômeno da palatalização é mais frequente na grande Aracaju. Este resultado pode ser decorrente de hipercorreção, o que pode ser melhor explorado em outras abordagens.

Referências

Câmara, J. M. (1970). *Estrutura da língua portuguesa*. Petrópolis, RJ: Editora Vozes.

Corrêa, T. R. d. A. (2019). *A variação na realização de /t/ e /d/ na comunidade de práticas da UFS: mobilidade e integração*. PhD thesis, Pós-Graduação em Letras, Universidade Federal de Sergipe.

Freitag, R. M. K. (2020). Effects of the linguistics processing: Palatals in brazilian portuguese and the sociolinguistic monitor. *University of Pennsylvania Working Papers in Linguistics*, 25(2):4.

Freitag, R. M. K. and Sá, J. J. D. S. (2019). Leitura em voz alta: variação linguística e o sucesso na aprendizagem inicial da leitura. *Ilha do Desterro A Journal of English Language, Literatures in English and Cultural Studies*, 72(3):41–62.

Freitag, R. M. K. and Santos, A. d. O. (2016). Percepção e atitudes linguísticas em relação às africadas pós-alveolares em sergipe. *A Fala Nordestina: entre a sociolinguística e a dialetologia. São Paulo: Blucher*, pages 109–122.

Freitag, R. M. K. and Souza, G. G. A. (2016). O caráter gradiente vs. discreto na palatalização de oclusivas em sergipe. *Tabuleiro de Letras*, 10(2):78–89.

Freitag, R. M. K., Souza Neto, A. F. d., and Corrêa, T. R. A. (2019). Panorama da palatalização em sergipe. *Língua e sociedade: diferentes perspectivas, fim comum*, page 63 – 80.

Kassambara, A. (2020a). *ggpubr: 'ggplot2' Based Publication Ready Plots*. R package version 0.4.0.

Kassambara, A. (2020b). *rstatix: Pipe-Friendly Framework for Basic Statistical Tests*. R package version 0.6.0.

Pinheiro, B. F. M., Silva, L. S., Araújo, L. C., Quirino, R. R., Souza, V. R. A., and Freitag, R. M. K. (2017). Processos fonológicos que passam da fala para a leitura. *Leitura, escrita e literatura: interseções e convergências. São Cristóvão, EdUFS*, pages 10–25.

Silva, L. S. (2021). *Análise acústica ou de oitiva: contribuições para o estudo da palatalização em Sergipe*. PhD thesis, Pós-Graduação em Letras, Universidade Federal de Sergipe.

Souza, G. G. A. (2016). *Palatalização de oclusivas alveolares em Sergipe*. PhD thesis, Pós-Graduação em Letras: Universidade Federal de Sergipe.

Souza, V. R. A., Silva, V. L. S., and de Araujo Júnior, M. M. (2020). Da fala à leitura. *Porto das Letras*, 6(1):167–199.

Souza Neto, A. F. d. (2008). *Realizações dos fonemas /t/ e /d/ em Aracaju, Sergipe.* São Cristóvão: EdUFS.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi,

K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

A propósito do verbo *falar* no português brasileiro: uma análise em *corpus* e em bases de dados verbais

Isaac Souza de Miranda Junior[1], Marcela Monteiro Lemos Couto[1 2], Francimeire Leme Coelho[1], Roana Rodrigues[3], Oto Vale[1]

[1] Universidade Federal de São Carlos (UFSCar), São Carlos, Brasil
[2] Instituto Federal de São Paulo (IFSP), Boituva, Brasil
[3] Universidade Federal de Sergipe (UFS), São Cristóvão, Brasil

isaacmiranda@estudante.ufscar.br, mmlcouto@gmail.com, flcoelho@estudante.ufscar.br, r.roanarodrigues@gmail.com, otovale@ufscar.br

Abstract. *This work investigates the syntactic-semantic proprieties of the verb "falar" (to speak) of Brazilian Portuguese (BP) in a journalistic corpus and three BP verbal databases (VerbNet.Br, VerboWeb and Verbo-Brasil). The data demonstrate a polyvalence and complexity of "falar". In addition, concerning the analysis of the databases, it was verified a need for manual reviews, made out by linguists, and/or expansions of the linguistic descriptions, especially when considering the relevance of constructions with the verb "falar" for different applications in the Natural Language Processing area.*

Resumo. *Neste trabalho, investiga-se o comportamento sintático-semântico do verbo "falar" no português brasileiro (PB) em um corpus jornalístico e em três bases de dados verbais do PB (VerbNet.Br, VerboWeb e Verbo-Brasil). Os dados demonstram a polivalência e complexidade de "falar". Além disso, no que se refere à análise das bases de dados, foi possível constatar a necessidade de revisões manuais, realizadas por linguistas, e/ou ampliações das descrições linguísticas, principalmente ao considerar a relevância de construções com o verbo "falar" para diferentes aplicações na área de Processamento de Língua Natural.*

1. Introdução

O verbo é um elemento nuclear das línguas naturais e atua, sobretudo, como responsável pela seleção dos argumentos necessários e essenciais para a construção de uma oração. Há muitas pesquisas descritivas sobre o comportamento dos verbos do português brasileiro, doravante PB, a partir de diferentes abordagens teórico-metodológicas. Pode-se mencionar trabalhos que se dedicam à descrição morfológica dessa classe de palavras [Bassani 2009], além de investigações de cunho sintático-semântico com propostas de tipologias [Rassi e Vale 2013] e a criação de variadas obras lexicográficas para os usuários comuns da língua [Fernandes 2005 (1940)], [Borba 1990], entre outros.

As descrições formalizadas sobre as construções verbais do PB podem ainda atuar como recursos para diversos empreendimentos na área de Processamento de Língua Natural (PLN), como a sumarização e tradução automáticas e a análise de sentimentos. Em um estudo recente [Rodrigues - no prelo], foram selecionadas, analisadas e

comparadas três bases de dados verbais do PB (VerbNet.Br, VerboWeb e Verbo-Brasil)[1], que, ademais de seu inegável valor descritivo em si, contribuem com os recursos disponíveis para o PLN. Em suma, o estudo enfatizou a existência de bases de dados robustas com informações sintático-semânticas sobre os verbos PB e o papel imprescindível do linguista na elaboração e revisão de dados lexicais para fins computacionais. Além disso, a pesquisa revelou a necessidade de adaptação, ampliação e, até mesmo, a criação de bases de dados verbais que considerem a polivalência e polissemia de determinados verbos e construções para além de seu uso como *pleno*, incluindo verbos auxiliares, verbos-suporte, ou verbos em expressões multipalavras.

Neste artigo, serão discutidos, mais especificamente, alguns dos aspectos sintático-semânticos do verbo *falar* no PB. Pode-se mencionar ao menos três motivações principais para a elaboração do presente estudo, a saber: (i) o fato de *falar* ser um verbo polivalente e bastante frequente na língua; (ii) o valor de *falar* em construções (diáteses) de comunicação [Couto 2017], que se constituem por um emissor, na posição de sujeito agente, uma mensagem e um destinatário, relevantes para o PLN ao se considerar, principalmente, sua atuação na introdução de discursos reportados que expressam *falas* (*opiniões*); e (iii) o comportamento de *falar* no PB, contrapondo-o ao verbo *dizer*, demarcando especificidades de usos se comparados a outras línguas naturais, como o espanhol [Humblé 2006] e o inglês [Dehaspe e Eynde 1991][2].

Portanto, esta pesquisa se dedica a discutir as seguintes questões: (i) quais os comportamentos sintático-semânticos de *falar* no PB, a partir de uma análise em *corpus*; (ii) como o verbo *falar* está descrito em três bases de dados verbais do PB: VerbNet.Br, VerboWeb e Verbo-Brasil; e (iii) quais informações sobre *falar* poderiam ser incluídas nas bases analisadas. Sendo assim, espera-se propor uma descrição sintático-semântica de *falar*, além de divulgar, analisar e contribuir com as bases de dados verbais disponíveis na língua.

Desse modo, este trabalho se organiza da seguinte maneira: na seção 2, são apresentados os usos de *falar*, encontrados no *corpus* de notícias do jornal Folha de São Paulo [Santana 2019], com a utilização do software Unitex. Na seção 3, descreve-se e compara-se o lexema *falar* nas três bases de dados selecionadas nesta investigação. Por fim, na seção 4, são apresentadas as considerações finais e os encaminhamentos para pesquisas futuras.

2. O verbo *falar* em *corpus*

A fim de verificar como o verbo *falar* se comporta em uso, além dos conhecimentos introspectivos dos pesquisadores e da classificação proposta por Borba (1990), foram analisadas as suas colocações no *corpus* do jornal Folha de São Paulo [Santana 2019], que consta de 167.053 notícias. Os dados foram processados pelo Unitex[3], uma

[1] As bases de dados selecionadas no trabalho - VerbNet.Br [Scarton 2013], VerboWeb [Cançado *et al*., 2018] e Verbo-Brasil [Duran *et al*. 2013] - sofreram atualizações na última década, estão disponíveis on-line e de maneira gratuita e apresentam a descrição sintático-semântica de, pelo menos, 1.000 lexemas verbais do PB.

[2] As pesquisas citadas destacam especificidades de uso de *hablar* e *decir*, em espanhol, e *to tell, to say, to speak* e *to talk*, em inglês, contrapondo-as aos casos em que *falar* atua como sinônimo de *dizer*, em alguns contextos do PB que introduzem um discurso reportado.

[3] Unitex/GramLab. Disponível em: <https://unitexgramlab.org/pt>. Acesso em: jun. de 2021.

ferramenta plurilingue, de código aberto, que possui dicionários eletrônicos e gramáticas locais para processamento e análise de dados textuais.

Quadro 1. Verbo *falar* em *corpus*

Colocações	**Especificidade**	**Estrutura**[4]	**Papéis Temáticos**	**Exemplo**
∅	Construção intransitiva	N_0 V	N_0 = Agente	*Camila **fala** tranquilamente.*
Preposição	sobre	N_0 V Prep Nnr	N_0 = Agente Nnr = Tópico	*O deputado não **fala** sobre o assunto*
	de			*Aquele cara só **fala** de economia.*
	em			*Dória **fala** em alterar a lei.*
	com[5]	N_0 V Prep NHum	N_0 = Agente NHum = Co-Agente	*Machado **fala** com a Folha.*
	a	N_0 V Prep NHum	N_0 = Agente NHum = Destinatário	*A candidata **falou** a uma multidão.*
	para			*Ele **fala** para uma plateia de 300 brasileiros.*
Complemento direto	Discurso reportado	N_0 V Que F_1 Prep $NHum_2$	N_0 = Agente F_1 = Mensagem $NHum_2$ = Destinatário	***Falei** para os jogadores que teríamos um desafio grande.*
Complemento direto	Nome restrito	N_0 V Nr	N_0 = Agente Nr = Objeto [língua]	*Ana **fala** (inglês, português, árabe)*
Complemento direto	Nome restrito	N_0 V Nr	N_0 = Agente Nr = Objeto [qualidade da mensagem]	*Rui **fala** (bobagem, palavrão, besteira, amenidades, inverdades).*
Expressões multipalavras	Diferentes estruturas	-	-	*Carlos **fala** (grego, pelos cotovelos, a mesma língua que Ana)*

[4] Notação da estrutura sintática: N_0, N_1 N_2: argumentos na posição de sujeito, primeiro complemento e segundo complemento; Nr: argumento preenchido por um nome restrito; Nnr: argumento preenchido por um nome sem restrição; NHum: argumento preenchido por nome humano; F_1: oração; Prep: preposição.

[5] Além da construção com a preposição *com*, foram encontrados, no *corpus*, construções pronominais com o verbo *falar* sempre construídas com sujeito plural. Neste trabalho, entende-se que tais construções são resultado de um processo transformacional de pronominalização, devido à simetria encontrada em: "A [não] fala com B" e "B [não] fala com A", assim "A e B [não] se falam", como no exemplo retirado do corpus: *Os irmãos não se falam desde o fim do Oasis.*

Ao todo, foram encontradas 49 mil ocorrências com *falar*[6] e investigadas 1.470 colocações. No Quadro 1, apresentam-se as construções analisadas, assim como uma proposta de descrição sintática (estrutura) e semântica (papéis temáticos)[7], seguindo a terminologia adotada pelo modelo teórico-metodológico do Léxico-Gramática [Gross 1975].

Conforme se verifica no Quadro 1, o verbo *falar* aparece em orações intransitivas, que, por sua vez, geralmente selecionam um circunstancial de modo. Além disso, encontraram-se casos em que *falar*: (i) seleciona uma preposição, desencadeando diferentes construções sintáticas e semânticas; (ii) seleciona complemento direto e indireto, constituindo um discurso reportado; (iii) seleciona um complemento direto com nome restrito (de idioma/dialeto), que, conforme aponta Borba (1990:740), se refere a uma construção estática, sinonímia de "ter a capacidade de"; (iv) seleciona um complemento com nome restrito, fazendo referência à qualidade da mensagem; e (v) é um constituinte nuclear de expressões multipalavras[8].

É interessante mencionar que, mesmo em se tratando de um *corpus* de textos jornalísticos, *falar* como verbo constituinte de construções comunicativas apresenta caráter informal, sendo encontrado em discursos reportados (marcado por aspas) e/ou em seções com menor formalidade na escrita, como o "caderno de esportes".

3. O verbo *falar* nas bases de dados verbais do PB

Conforme mencionado, nesta seção, serão apresentadas as descrições de *falar*, juntamente com uma análise comparativa com o Quadro 1, na VerbNet.Br, Verbo-Brasil e VerboWeb, bases de dados verbais relevantes do PB, que podem ser utilizadas em diferentes empreendimentos na área de PLN.

A VerbNet.Br [Scarton 2013] é um recurso léxico-computacional (RLC) que agrupa verbos do PB em diferentes classes semânticas. O recurso foi criado de maneira semiautomática a partir do RLC do inglês, a VerbNet [Schuler 2005]. As classes semânticas, inspiradas em Levin (1993), agrupam os verbos de acordo com semelhanças semânticas compartilhadas e suas alternâncias sintáticas.

Por conta da abordagem inter-linguística utilizada na criação desse RLC, as classes presentes na VerbNet.Br são alinhadas diretamente com as classes do inglês. Cada uma delas apresenta as seguintes informações: (i) os membros que a compõem; (ii) os papéis temáticos, que foram herdados da versão do inglês; (iii) as restrições seletivas, que são impostas aos papéis temáticos (como animacidade, concretude etc.), também herdadas do inglês; os frames sintáticos do PB e do inglês, que descrevem a transitividade verbal e, também, itens lexicais selecionados em alternâncias em particular; e (iv) os predicados semânticos, que fornecem informações sobre as relações entre os participantes e o evento da ação verbal.

[6] Este número abarcou também as formas nominais *a fala* e *o falar*. Neste trabalho, analisaram-se somente as construções verbais.

[7] Os papéis temáticos utilizados na pesquisa baseiam-se no estudo de Santos (2014).

[8] Em Vale (2002) estão descritas e classificadas, sintático-semanticamente, 22 expressões multipalavras com o verbo *falar* no PB, tais como: *falar da boca pra fora* ou *falar de barriga cheia*.

Quadro 2. Verbo *falar* na VerbNet.Br

FALAR		
Classes	**Papéis temáticos [restrições seletivas]**	**Alternâncias**
amuse-31.1	Experiencer [+animate] Stimulus Result	V; V_NP; V_NP_PP [com]
chit_chat-37.6	Agent [+animate | +organization] Co-Agent [+animate | +organization] Topic [+communication]	V; V_PP[com]; V_PP[sobre]; V_PP[sobre]_PP[com]
correspond-36.1	Agent [+animate | +organization] Co-Agent [+animate | +organization] Theme	V; V_PP[com]_PP[sobre]; V_PP[sobre]
dub-29.3	Agent [+animate | +organization] Theme [+concrete | +organization] Result	V_NP
establish-55.5	Agent [+animate | +organization] Theme	V_NP
lecture-37.11	Agent [+animate | +organization] Topic Recipient [+animate | +organization]	all
talk-37.5	Agent [+animate | +organization] Co-Agent [+animate | +organization] Topic [+communication]	V; V_PP[com]; V_PP[com]_PP[sobre]; V_PP[para]; V_PP[para]_PP[sobre]; V_PP[sobre]_PP[com]; V_PP[sobre]_PP[de]; V_PP[sobre]_PP[para]
transfer_mesg-37.1.1	Agent [+animate | +organization] Topic Recipient [+animate | +organization] Source	V; V_NP; V_NP_PP[a]; V_NP_PP[para]

Nota. Fonte: Adaptado de Scarton (2013).

Conforme se observa no Quadro 2, o verbo *falar* está presente em oito classes da VerbNet.Br. e muitos *frames* sintáticos estão presentes em mais de uma classe. É importante salientar que a VerbNet.Br não exemplifica as alternâncias com dados reais do PB ou mesmo criados por introspecção do falante, o que dificulta, em alguns casos, a interpretação dos dados descritos. Para a alternância sintática **V_PP[sobre]_PP[de]**, da classe *talk-37.5*, por exemplo, não foi localizada nenhuma ocorrência compatível no *corpus* de dados jornalísticos como se verifica nos dados dispostos no Quadro 1. A pista para o PB é que se trata de um *frame* com duas preposições (*sobre/de*) e com a restrição seletiva [+*comunication*] imposta ao papel temático Topic. Note-se que as preposições *sobre* e *de* precedidas do verbo *falar* selecionam potencialmente argumentos com papel temático assunto (Tópico) ou *conteúdo da mensagem* (Mensagem), conforme defende Couto (2017). No entanto, no *corpus* não há ocorrência do verbo *falar* com essas duas preposições antecedendo o *conteúdo da mensagem*.

Salienta-se, ainda, o comportamento do verbo *falar* descrito na classe *lecture-37.11*, o qual possui a descrição *all* no que diz respeito às alternâncias sintáticas. Isso quer dizer que o verbo *falar* admite todas as alternâncias sintáticas do PB. Essa é uma

informação difícil de ser atestada, porque não existe uma delimitação teórico-metodológica de quais seriam essas alternâncias ou uma lista das alternâncias do PB para realizar buscas e testes com maior rigor metodológico. Novamente, a ausência de exemplos é um outro grande limitador dessa verificação.

A VerboWeb [Cançado *et al*. 2018] é uma base digital de consulta lexical, que, devido sua organização e metodologia, pode ser utilizada como RLC. Nela, os verbos estão classificados em função de suas características sintático-semânticas também com base nas classes de Levin (1993). A descrição proposta foi realizada de maneira manual por linguistas e a tipologia verbal ocorre em duas etapas de categorização: classes e subclasses. A classificação em *classes* consiste em uma divisão dos verbos em função de quatro características: *Estrutura sintática básica* (disposição sintática dos constituintes: Sintagma Nominal+Verbo, etc.); *Papéis temáticos* (relação semântica entre um predicado e seus argumentos: agente, paciente, etc.); *Decomposição de predicado* (demonstração de um predicado em função de predicados primitivos, ou seja, a descrição dos predicados irredutíveis que compõem um predicado complexo: *O palestrante falava muito* ⇒ [X DO <EVENT>]); *Aspecto lexical* (classificação dos predicados em função de duração, telicidade e dinamicidade nas classes de Vendler (1967)). Por sua vez, a organização das subclasses consiste no arranjo interno dos verbos de uma classe em função de suas características sintático-semânticas em comum.

Quadro 3. Verbo *falar* na VerboWeb

Verbo: *Falar* **Exemplo:** *O palestrante falava muito*
Classe: Atividade: verbos internamente causados (inergativos) **Propriedades da Classe:** Conteúdo semântico recorrente na classe: x faz/produz um evento em si mesmo – Estrutura sintática básica: [SN V] (verbo intransitivo) – Estrutura de papéis temáticos: {Agente} – Estrutura de decomposição de predicados: [X DO <EVENT>] – Aspecto lexical básico: atividade – Licencia um objeto cognato: *O palestrante falou uma fala bonita*. – Licencia um adjunto equivalente ao objeto cognato: *O palestrante falou muito bonito*.
Subclasse: Verbos de expressão (modo de fala) **Propriedades da Subclasse:** Denota um evento de fala – Licencia mensagem comunicada na posição de objeto: *O palestrante falava muita besteira*. – Licencia a mensagem e o destinatário nas posições de objeto: *O palestrante falava muita besteira para o público jovem*. – Licencia um objeto sentencial : *O palestrante falava que os jovens são o futuro do país*. – Licencia destinatário na posição de objeto indireto: *O palestrante falava animado para o público jovem*.

Nota. Fonte: Cançado M., Amaral, L., e Meireles, L. (2021). VerboWeb. Falar. http://www.letras.ufmg.br/sistemas/verboweb_cliente/ver_verbo.php?id=1220, junho de 2021.

O verbo *falar* é descrito dentro da VerboWeb como parte da subclasse dos *verbos de expressão* dentro da classe *verbos de atividade* internamente causados (inergativos), como é replicado no Quadro 3.

Com o Quadro 3, pode-se notar que a base descreve dois usos de *falar*, o primeiro como verbo intransitivo, fazendo parte da classe dos verbos inergativos, e o segundo como verbo de comunicação, referente à subclasse dos verbos de expressão, desconsiderando, no entanto, os casos em que *falar* ocorre precedido das preposições *sobre, de*, *em* e *com*, presentes no Quadro 1.

Ademais, pode-se reparar também que o "licenciamento de mensagem comunicada na posição de objeto" proposto na VerboWeb faz referência ao mesmo uso (iv) no Quadro 1, porém, o exemplo utilizado (*O palestrante falava muita besteira*) não se refere ao *conteúdo da mensagem* como proposto na VerboWeb, mas sim, como descrito na seção 2, à *qualidade da mensagem*. *Besteira* no exemplo, não seria a fala em discurso reportado, mas sim uma qualidade/característica associada a ele pelo locutor.

Por último, a Verbo-Brasil [Duran *et al*. 2013] é um RLC em que, ao contrário das bases anteriores, os verbos não estão distribuídos em classes maiores de categorização, mas sim descritos individualmente em função dos diferentes sentidos (*roleset*) que podem assumir. Mesmo que exista dentro da base uma possível classificação (*vncls*: *verbnet class*) em função das classes da VerbNet [Schuler 2005], existem sentidos que são vazios quanto essa classificação por não estarem contemplados na VerbNet.

Assim como as outras bases, a Verbo-Brasil é orientada pela descrição das classes de Levin (1993), em que cada verbo é classificado de acordo com suas propriedades sintáticas e aspectos semânticos em comum. Além disso, na Verbo-Brasil, cada sentido é descrito e classificado em função dos seus argumentos e vinculado a um sentido existente no PropBank [Palmer *et al*. 2005] quando possível. Cada argumento recebe uma identificação de papel temático (*vnrole*: *verbnet role*) em função dos papéis temáticos descritos na VerbNet [Schuler 2005].

O verbo *falar* é descrito dentro da Verbo-Brasil [Duran *et al*. 2013] com apenas um sentido contendo quatro argumentos que não necessariamente precisam ocorrer juntos em uma sentença. Dentro da descrição de *falar*, encontra-se também o sentido de duas expressões multipalavras: *falar mal* e *dar o que falar* e os seus respectivos argumentos e papéis temáticos. A descrição do sentido de *falar* pode ser verificada no Quadro 4:

Quadro 4. Verbo *falar* na Verbo-Brasil

	Sentido	**Estrutura Argumental**
Falar	**Roleset id**: falar.01, dizer; declarar; **vncls:** 37.7; **Mapeamento para o inglês:** say.01, talk.01, speak.01	**Arg0**: *falante* (vnrole: 37.7-agent) **Arg1**: *idioma falado; assunto (falar de, falar sobre, falar contra, falar a favor de) ou elocução* (vnrole: 37.7-topic) **Arg2**: *ouvinte, interlocutor (falar a, falar para, falar com)* (vnrole: 37.7-recipient) **Arg3**: *atributo do arg0 (falar como)*[9]

Nota. Fonte: Duran, M. S., Martins J. P., Coimbra, M., Patire, P. A., Hartmann N., e Aluísio, S. (2014). Verbo-Brasil. Framefile-falar-v. http://143.107.183.175:21380/verbobrasil/textoFrames/falar-v.html, junho de 2021.

[9] A construção *falar como* foi encontrada no corpus mas, por não se tratar de um argumento selecionado pelo verbo, não foi inserida no Quadro 1.

A forma como *falar* está descrito no Quadro 4, mesmo que abrangente, pode gerar alguns problemas de descrição, uma vez que o **roleset** *falar.01* apresenta 32 exemplos[10] diferentes de construções com o verbo, em que cada construção tem especificidades sintáticas distintas, apresentando números distintos de argumentos. Vale ressaltar também que em alguns exemplos da Verbo-Brasil, como o **roleset** *falar*.01 exemplo 2 (*Não ouvi falar nada sobre arbitragens*) e o **roleset** *falar*.01 exemplo 30 (*Takuo Hirano e Tetsuyuki Hirano, pai e filho, vieram falar sobre o novo conceito de design que estão desenvolvendo*), são representados na base com estruturas argumentais idênticas (contando apenas com ***arg1***), o que é um problema considerando que ambos os exemplos apresentam estruturas distintas, enquanto que no exemplo 2 não há a presença explícita de um ***arg0***, no exemplo 30 *Takuo Hirano e Tetsuyuki Hirano* é **arg0** não somente de *vir*, mas como também de *falar*.

Como visto na seção, cada base exprime uma cobertura distinta dos usos de *falar*, isso ocorre devido à polivalência e polissemia de *falar*, reforçando a necessidade da revisão e ampliação das descrições disponíveis sobre esse verbo nas bases de dados analisadas.

4. Considerações finais

Por se tratar de um verbo bastante frequente, o comportamento do verbo *falar* apresenta um número considerável de empregos que merecem ser considerados em qualquer trabalho descritivo. Esse é um fato comum em itens de alta frequência que apresentam, via de regra, uma polissemia que nem sempre é detectável à primeira vista.

De fato, retomando as questões que nortearam esta pesquisa, nota-se que o verbo *falar* no PB apresenta um comportamento sintático-semântico variado, que requer estudos aprofundados. Além dos casos recenseados no Quadro 1, destaca-se ainda a necessidade de análises dedicadas à polivalência e polissemia de *falar* em construções como: *falar em* (*seu discurso + casamento + inglês...*), *falar* (*como + na condição de + na qualidade de*) (*presidente + professor*) e *falar* (*por* + no lugar de) *alguém*. Tais complementos poderiam ser atribuídos à estrutura argumental do verbo? Esse tipo de pergunta poderia ser estendida a outras construções de comunicação, considerando-se o comportamento de diferentes *verba dicendi* [Costa e Freitas 2017] [Baptista 2010].

No que se refere ao estudo de *falar* nas bases de dados verbais, verificou-se que todas as bases enfatizam o uso de *falar* em construções de comunicação. Constatou-se ainda que as bases verbais construídas para fins computacionais (VerbNet.Br e Verbo-Brasil) necessitam de uma revisão linguística especializada, seja para incluir exemplos reais da língua, seja para refinar a descrição e propor classificações mais granulares dos dados. Por sua vez, a base que se constrói sob descrições linguísticas manuais (VerboWeb) parece ser a que possui maior coerência teórico-metodológica, embora apresente um número reduzido de construções analisadas, podendo ampliar o estudo de tal verbo em futuras atualizações da base.

Sendo assim, acredita-se que as informações sobre o verbo *falar* levantadas no *corpus* e descritas neste trabalho podem impulsionar não só ao incremento e atualização das bases de dados verbais elencadas, como também à realização de novos estudos sintático-semânticos sobre esse verbo.

[10] Os exemplos da Verbo-Brasil foram extraídos do *corpus* PLN-Br [Bruckschen *et al*. 2008].

Agradecimentos

Os autores agradecem à CAPES (Código Financeiro 001) e ao Centro de Inteligência Artificial (C4AI) da Universidade de São Paulo, apoiado pela IBM e FAPESP (nº 2019/07665-4).

Referências

Baptista, J. (2010). Verba dicendi: a structure looking for verbs. In: Nakamura, T., Laporte, E., Dister, A., Fairon, C. *Les Tables – La grammaire du français par le menu*. Mélanges en hommage à Christian Leclère. Cahiers du CENTAL, 6, 11-20.

Bassani, I. S. (2009). *Formação e interpretação dos verbos denominais do português do Brasil*. Dissertação (Mestrado em Semiótica e Linguística). São Paulo, Universidade de São Paulo.

Borba, F. S. (coord.). (1990). *Dicionário gramatical de verbos do português contemporâneo do Brasil*. São Paulo: Editora UNESP.

Bruckschen, M., Muniz, F., Souza, J. G. C., Fuchs, J. T., Infante, K., Muniz, M. e Aluísio, S. M. (2008). Anotação linguística em XML do corpus PLN-BR. *Série de relatórios do NILC*, NILC-ICMC–USP.

Cançado, M., Amaral, L., e Meirelles, L. L. (2018). Verboweb: Uma proposta de classificação verbal. *Revista da Anpoll*, 1(46), 123-141.

Cançado M., Amaral, L., e Meireles, L. (2021). VerboWeb. Falar. http://www.letras.ufmg.br/sistemas/verboweb_cliente/ver_verbo.php?id=1220, Junho de 2021.

Costa, B. F. S., e Freitas, C. (2017). Verbos de elocução em português: um estudo descritivo com base em grandes corpora e motivado pela linguística computacional. *Fórum Linguístico*, 14(3), 2266-2285.

Couto, M. M. L. (2017). *O estudo das valências verbais aplicado às construções de comunicação do português brasileiro*. Dissertação (Mestrado em Estudos Linguísticos). Belo Horizonte, Universidade Federal de Minas Gerais.

Dehaspe, L. e van den Eynde, K. (2012). The Pronominal Approach to Verbal Valency: A formal description of speak, say, tell, and talk. In *Betriebslinguistik und Linguistikbetrieb* (pp. 273-280). Max Niemeyer Verlag.

Duran, M. S., Martins, J. P., e Aluisio, S. M. (2013). Um repositório de verbos para a anotação de papéis semânticos disponível na web. In: *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*. <https://www.aclweb.org/anthology/W13-4820.pdf>, acesso em junho de 2021.

Duran, M. S, Martins J. P., Coimbra, M., Patire, P. A., Hartmann N. e Aluísio, S. (2014). Verbo-Brasil. Framefile-falar-v. http://143.107.183.175:21380/verbobrasil/textoFrames/falar-v.html>, acesso em junho de 2021.

Fernandes, F. (2005 [1940]). *Dicionário de verbos e regimes*. 45 ed. Porto Alegre: Globo.

Gross, M. (1975). *Méthodes en Syntaxe*: régime des constructions complétives. Paris: Hermann,

Humblé, P. (2006). Falsos cognados. Falsos problemas. Un aspecto de la enseñanza del español en Brasil, *Revista de Lexicografía*, 2005-2006, 12: 197-207 <https://ruc.udc.es/dspace/handle/2183/5510>, acesso em junho de 2021.

Levin, B. (1993). *English Verb Classes And Alternations*: A Preliminary Investigation. Chicago: University of Chicago Press.

Palmer, M., Gildea, D., e Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1), 71-106.

Rassi, A. P. e Vale, O. A. (2013). Tipologia das construções verbais em PB: uma proposta de classificação do verbo dar. *Caligrama*, Belo Horizonte, v. 18, n. 2.

Rodrigues, R., Lemos-Couto, M. M., Coelho, F. L., Miranda Jr, I. e Vale, O. (2021) Bases de dados verbais do português brasileiro (artigo em vias submissão).

Santana, M. (2019). "News of the Brazilian Newspaper - 167.053 news of the site Folha de São Paulo (Brazilian Newspaper)", <https://www.kaggle.com/marlesson/news-of-the-site- -folhauol>, acesso em junho de 2021.

Santos, R. P. T. (2014). *Automatic Semantic Role Labeling for European Portuguese*. Dissertação (Mestrado em Ciências da Linguagem). Universidade do Algarve, Faro.

Scarton, C. E. (2013). *VerbNet.Br: construção semiautomática de um léxico verbal online e independente de domínio para o português do Brasil*. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional). Universidade de São Paulo, São Carlos.

Schuler, K. K. (2005). *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

Vale, O. A. (2002). *Expressões cristalizadas do português do Brasil: uma proposta de tipologia*. Tese (Doutorado em Linguística e Língua Portuguesa). Universidade Estadual Paulista Júlio de Mesquita Filho, Araraquara.

Vendler, Z. (1967). *Linguistics in Philosophy*. Ithaca, NY: Cornell Univ. Press.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Provérbios portugueses usuais: distribuição em corpora

Sónia Reis[1], **Jorge Baptista**[1,2], **Nuno Mamede**[1,3]

[1]HLT – Human Language Technologies Lab
Lisboa, Portugal

[2]Universidade do Algarve, Faculdade de Ciências Humanas e Sociais
Faro, Portugal

[3]INESC ID Lisboa
Lisboa, Portugal

reis.soniamm@gmail.com, jbaptis@ualg.pt, nuno.mamede@inesc-id.pt

Abstract. *Proverbs are a special type of linguistic unit that have been largely ignored by the Natural Language Processing (NLP) community, though they pose interesting challenges to NLP systems. This paper presents the procedure of integrating the Paremiological Minimum of Portuguese into the STRING system, and the distribution of these most usual proverbs in three different (European) Portuguese corpora.*

Resumo. *Os provérbios são um tipo especial de unidades linguísticas que tem sido amplamente ignorado pela comunidade de Processamento de Linguagem Natural (PLN), apesar de levantarem desafios interessantes para o processamento. Este artigo apresenta o procedimento de integração do Mínimo Paremiológico do Português no sistema STRING e a distribuição desses provérbios mais usuais em três corpora distintos do português (europeu).*

1. Processamento de provérbios em textos

Tanto quanto sabemos, os provérbios estão praticamente ausentes dos recursos linguísticos de muitos sistemas de Processamento de Linguagem Natural (PNL) desenvolvidos para o português. Tal resulta, provavelmente, do seu estatuto linguístico pouco consensual, algures entre o léxico e a cultura, que está certamente na origem da sua ausência dos dicionários de língua, encontrando-se antes recenseados em coletâneas de provérbios, onde convivem com outras expressões (locuções) de natureza muito variada, frequentemente idiomáticas (i.e. semanticamente não composicionais). Outro aspeto é o seu extenso número e a sua variação formal, lexical e sintática, que dificultam o reconhecimento destas expressões em textos [Rassi et al. 2014a, Rassi et al. 2014b]. Finalmente, o seu estatuto de *citação*, nem sempre formalmente assinalado (aspas/discurso relatado), confere-lhe uma certa autonomia relativamente o texto em que se insere, permitindo que sejam violados mecanismos de coesão discursiva (processos anafóricos, concordância verbal na subordinação), ainda que os seus elementos se prestem a ser retomados noutros locais do discurso, dando origem a criativos jogos de palavras. São, pois, complexos os desafios que os provérbios levantam ao seu processamento automático em textos.

O estudo dos provérbios no quadro do PLN pode dar origem a aplicações societalmente relevantes. [Reis and Baptista 2016b] desenvolveram dois conjuntos de jogos

com provérbios, linguisticamente motivados, com base nos provérbios do MP, que podem ser utilizados para ensino de língua ou até para o dignóstico/terapia de algumas patologias da linguagem. Mais recentemente, [Mendes and Oliveira 2020a] testaram diferentes técnicas de representação semântica (e.g. Jaccard) para avaliar a similaridade semântica entre um *corpus* de aproximadamente 1.600 provérbios e manchetes de jornais, no quadro de uma tarefa de recomendação automática de texto. A avaliação dos resultados por meio de questionário revelou que, na maioria das vezes, as pessoas conseguiram estabelecer uma relação entre a expressão selecionada e a manchete correspondente, chegando, mesmo, a achá-la potencialmente engraçada. Verificam ainda que os provérbios que partilham as mesmas palavras com a manchete, nomeadamente os escolhidos por métodos mais simples (e.g. Jaccard) são mais facilmente relacionados com as manchetes. Já o uso de representações semânticas mais profundas como *word embeddings* (e.g. BERT) [Mendes and Oliveira 2020b] revelou piores resultados, o que foi justificado pelos autores pela linguagem figurada própria destas expressões. Numa linha mais próxima da deste artigo, [Davis et al. 2021] analisam a frequência e a dinâmica de provérbios em diferentes tipos de texto e ao longo do tempo, baseando-se numa lista de +14 mil provérbios americanos e usando como *corpora*: (i) o *corpus* Gutenberg (60.000 documentos); (ii) o *corpus* New York Times (1,8 milhões de artigos de 1987-2007); (iii) dados do Google (a partir de 2020); e (iv) dados do Twitter (a partir de janeiro de 2021). Os autores identificaram as expressões que mais se repetem em cada um destes *corpora*, admitindo que uma limitação do seu estudo é a questão da representatividade, já que os dados de partida, de um dicionário de provérbios americano, são limitados. Ainda assim, a maior disponibilidade de dados textuais abre caminho a estudos fraseológicos longitudinais.

Enquanto unidade linguística, dada a sua variação, os provérbios podem ser organizados em *unidades paremiológicas* (UP), isto é, unidades conceptuais representadas pelo conjunto das múltiplas variantes de um mesmo provérbio e que são definidas com base em critérios formais, semânticos e pragmáticos [Reis 2020]. Dado o elevado número de provérbios conhecidos, na ordem das dezenas de milhar, pode ser útil restringir o âmbito da investigação ao *Mínimo Paremiológico* (MP) de uma língua, ou seja, o conjunto dos provérbios mais usuais de uma dada língua, conhecidos e empregues pela maioria dos falantes de uma dada comunidade linguística.

Até à data, os provérbios têm estado fora do âmbito dos objetos linguísticos processados pela STRING [Mamede et al. 2012], uma cadeia de Processamento de Linguagem Natural, construída especificamente para processar textos em português. Este artigo apresenta a integração na STRING do conjunto de expressões que formam o *Mínimo Paremiológico do Português Europeu* [Reis and Baptista 2020], e está organizado do seguinte modo: na secção seguinte, apresentamos o mínimo paremiológico e a forma como foi constituído; depois, descrevemos o método de representação dos provérbios do MP e suas variantes para a sua integração na STRING; de seguida, descrevemos os 3 *corpora* utilizados neste estudo e a distribuição dos provérbios do MP nesses textos, procurando determinar fatores que expliquem as assimetrias encontradas; o artigo conclui-se com breves notas sobre perspetivas futuras.

2. Mínimo paremiológico

O *Mínimo Paremiológico do Português Europeu* (MP) [Reis and Baptista 2020] foi determinado, reunindo primeiro várias listas de provérbios a partir de dicionários/coletâneas

de referência [Moreira 1996, Costa 1999, Parente 2005, Machado 2011] e depois estimando a sua *disponibilidade lexical* com base na sua ocorrência em variadas fontes [Reis and Baptista 2016a]. Em primeiro lugar, com base numa listagem de +114 mil provérbios foi produzida, de forma independente, uma classificação manual por dois anotadores humanos, que identificaram os provérbios que consideravam serem "usuais", sendo depois comparadas as respetivas anotações. Os resultados obtidos permitiram atribuir um nível provisório de disponibilidade lexical aos provérbios, utilizando uma escala de 3 níveis (0, 1 e 2, em que '0' corresponde a expressões pouco usuais, raramente disponíveis; '1', moderadamente disponíveis; e '2' para as expressões muito usuais, altamente disponíveis). De seguida, a partir de uma seleção aleatória de provérbios de cada um dos níveis assim determinados, foi aplicado um questionário *on-line*, a que responderam 735 informantes, o qual veio confirmar, em grande medida, a seleção dos anotadores. Por outro lado, foi determinada a frequência dessa mesma lista de provérbios obtida a partir da *web*, no domínio de topo `.pt` e utilizando dois motores de busca (Bing e Google). Os resultados de ambos os motores de busca foram muito semelhantes (Pearson: 0.96), e mostram uma correlação relativamente alta (Pearson: aprox. 0.70) com a classificação manual dos anotadores. A mesma experiência foi realizada no *corpus* CETEM*Público* [Santos and Rocha 2001] e os resultados obtidos correlacionaram-se bem com a classificação manual dos anotadores, já que não há provérbios de nível 0 (raros ou pouco usuais) e que foram encontrados mais casos de provérbios de nível 2 (muito usuais) do que de nível 1 (usuais). Uma vez que os provérbios são frequentemente utilizados num contexto educacional para a aprendizagem de língua (e cultura), foi reproduzida a experiência tanto em manuais didáticos de português L1 [Reis and Baptista 2016c] como em manuais de português L2 [Reis and Baptista 2017]. Os resultados destas experiências tornaram possível o estabelecimento do *Mínimo Paremiológico do Português Europeu*, que contém 318 UPs, ou seja os provérbios mais usuais e representando mais de 14.500 expressões proverbiais (variantes) [Reis and Baptista 2020]. Por outras palavras, cada provérbio está associado ao conjunto das suas variantes, já recolhidas em dicionários e coletâneas especializadas e, em alguns casos, variantes encontradas em outras fontes, incluindo a Internet.

3. Expressões regulares e integração do MP na STRING

Descrevemos agora a metodologia de integração do léxico do Mínimo Paremiológico no sistema de processamento computacional do português STRING [Mamede et al. 2012], com vista à identificação destes provérbios nos textos em que ocorram. Para tal, determinaram-se primeiro os requisitos de informação necessários para encontrar a solução ótima com vista à integração deste tipo de objetos na arquitetura geral da STRING. Um dos requisitos fundamentais é poder representar os provérbios contando com a análise morfossintática dos itens lexicais, no mínimo, a informação quanto ao lema das formas e a respetiva categoria morfossintática. Outro requisito prende-se com a possibilidade de inserção de separadores no meio das expressões, já que a pontuação de muitas expressões proverbiais é bastante "criativa". Assumiu-se, além disso, que, à semelhança de uma palavra composta, os provérbios funcionam como uma "ilha" no corpo do texto, não sendo relevante (pelo menos numa análise inicial) atribuir-lhes uma análise sintática, proceder à extração de papéis semânticos [Talhadas et al. 2013], ou mesmo fazer a resolução de relações anafóricas [Mitkov 2002, Marques 2013]. Finalmente, ainda que nesta fase apenas se tenha representado os provérbios do Mínimo

Paremiológico, constituído pelas 318 unidades paremiológicas mais usuais e frequentes, pretende-se que o método permita, progressiva e cumulativamente, representar uma mais lata extensão do rico património linguístico e cultural que constitui o léxico dos provérbios (dezenas de milhar de unidades paremiológicas).

Assim, considerando a arquitetura da STRING, decidiu-se delimitar e anotar o provérbio numa fase bastante inicial do processamento, tratando-o como uma unidade textual, mas só depois da fase de análise e desambiguação morfossintática, para poder recorrer a essa informação. Isso corresponde a utilizar o mecanismo que, no analisador sintático da STRING, se chama de *gramáticas locais* (*GramLoc*). Estas consistem em, numa fase preliminar de análise sintática (ing. *parsing*), construir um nó `noun` na estrutura da frase, com base no emparelhamento de uma dada expressão com um padrão descrito por uma *expressão regular* (*RegExp*), associando-lhe depois as propriedades linguísticas relevantes. No sistema, as expressões regulares das gramáticas locais obedecem a uma sintaxe própria, cujo formalismo é razoavelmente complexo, pelo que a sua construção manual, diretamente a partir das formas a descrever, está muito sujeita à introdução de erros. Por essa razão, a integração dos provérbios na STRING foi feita em duas etapas: num primeiro momento, as variantes dos provérbios são representadas por meio de expressões regulares relativamente transparentes, cuja construção é mais simples e mais adequada para um linguista, preocupado sobretudo em descrever os padrões combinatórios dos provérbios, do que o formalismo usado pelo sistema nas suas gramáticas locais; e num segundo momento, um programa especialmente construído para o efeito converte essas expressões regulares no formalismo mais complexo das gramáticas locais usado pela STRING.

Dada a natureza da variação formal observada neste tipo de expressões, adotou-se uma representação por expressões regulares que permitisse descrever a estrutura e variação destas formas tendo em conta: (i) possibilidade de indicação dos items lexicais tanto pela *forma* como pelo <*lema*>, e.g. `gato` ou `<gato>`, eventualmente acompanhada da categoria morfossintática, e.g. `<ser,V>`; (ii) possibilidade de representar qualquer sequência de separadores (‘`#`’) ou de palavras (‘`@`’); (iii) operadores mais usuais de expressões regulares como a disjunção ‘`|`’ ou a sequência vazia ‘`$`’. Assim, por exemplo, considerando o provérbio *Velhos são os trapos* (MP-315), em que se observa a possibilidade de comutar *trapos* com *farrapos*, o linguista constrói a RegExp:

```
velhos <ser,V> os (trapos|farrapos).
```

que é depois convertida automaticamente no formalismo da GramLoc:

```
1> noun[proverb=+]
@= ?[surface:velhos];?[surface:Velhos],
verb[lemma:ser], ?[surface:os],
?[surface:trapos];?[surface:farrapos].
```

Esta GramLoc constrói um nó `noun`, com a propriedade `proverb=+`, juntando a sequência de palavras formadas pelas formas *velhos*, uma forma flexionada associada ao lema do verbo *ser*, o artigo *os* e as formas em alternativa (`;`) *trapos* ou *farrapos*; note-se que a primeira forma tem de ser representada como duas alternativas, para dar conta do emprego de maiúscula inicial (uma alternativa seria usar o lema `<velho>`). Num segundo momento, ao nível da análise sintática do texto, uma regra geral extrai a

dependência unária PROVERB, que identifica o provérbio:

```
| noun#1[proverb] | PROVERB(#1).
```

O sistema identifica, assim, este provérbio nos textos em que ocorra, produzindo como saída a dependência correspondente (exemplo retirado do *corpus* Desportivo, v. adiante; na dependência os elementos do provérbio são, para já, identificados pelo respetivo lema):

> **[DSP]** *Cumpriu o seu 300º jogo para o campeonato e continuou a provar que* velhos são os trapos.
> `PROVERB(velho ser o trapo)`

Para cada unidade paremiológica do MP foram construídas manualmente mais de 1.100 RegExp, dando conta de fenómenos tão variados como permutas, redução de elementos, variação lexical e inserção de sinais de pontuação. Em paralelo, foram produzidos semiautomaticamente mais de 14.500 exemplos ilustrativos das variantes representadas por essas RegExp, que funcionam como material de validação das GramLoc que são construídas a partir daquelas.

4. Distribuição dos provérbios do MP em *corpora*

O estudo da distribuição dos provérbios do *Mínimo Paremiológico do Português Europeu* (MP) em *corpora* visa não só avaliar o desempenho da STRING em textos de natureza e tópicos variados, como também procurar padrões que revelem diferenças no uso destas expressões em contextos diferentes. Para aferir a distribuição em *corpora* dos provérbios usuais do MP, foram utilizados 3 *corpora* já existentes e processados pela STRING: o *CETEMPúblico* (CTP) [Santos and Rocha 2001], o *Desportivo* (DSP) e o *Parlamento* (PRL) [Trindade 2020]. O *corpus* CETEMPúblico, de cariz jornalístico, é um *corpus*, disponível publicamente, que contém 175.350.145 palavras (após processamento na STRING) e é distribuído pela Linguateca[1]. O *corpus Parlamento* resulta da compilação das atas de sessões da Assembleia da República, de 1976 a 2018, apresentando 123.633.859 palavras. Finalmente, o *corpus Desportivo*, com 100.161.374 palavras, é um *corpus* de texto jornalístico, de temática desportiva (sobretudo futebol), composto por textos dos jornais *O Jogo* (1999-2005) e *A Bola* (2000-2006).

Através da aplicação das gramáticas locais da STRING aos 3 *corpora*, foram identificadas 7.334 ocorrências de provérbios, das quais, 45,1% encontram-se no Desportivo, 30,4% no CETEMPúblico e 24,5% no Parlamento. Por razões de espaço, apenas alguns resultados serão reportados neste artigo[2].

A grande maioria das expressões encontradas nos *corpora* correspondem efetivamente a instâncias dos provérbios do MP, sendo a precisão alcançada bastante elevada (99,86%). A avaliação da abrangência, pelo menos para já, só faria sentido para os provérbios do MP, já formalizados, e respetivas variantes. Um pequeno número de casos (#10) são, porém, *falsos-positivos* e correspondem às seguintes situações: (i) *inserções*, como sucede na RegExp: `($|a) paciência @ <ter,V> limites.` que descreve o

[1] `www.linguateca.pt/cetempublico`

[2] A lista integral das UP, com a distribuição dos provérbios por cada *corpus*, bem como outras informações quanto às respetivas variantes, encontra-se disponível *on-line*: https://www.researchgate.net/publication/354997837; DOI: 10.13140/RG.2.2.14732.44164.

provérbio **MP_024** *A paciência tem limites*, pelo que o sistema emparelha esta expressão com a frase seguinte:

[CTP] [. . .] *José Afonso mostra que a festa de_os sons não* **tem limites** .

(ii) expressões ambíguas como, por exemplo:

[CTP] *Só que o problema não é* **querer , é poder**.
[CTP] *Deve situar se acima de os partidos políticos , sem* **querer ser contrapoder** *nem em relação a o Governo , nem em relação a a Assembleia da República*
[CTP] *E , sugere , o ' salve se* **quem puder ' poderá** *não ser assim tão mau .*
[DSP] *E a deslocação a o American Airlines Arena não será o melhor jogo para* **quem procura encontrar** *definitivamente o caminho de as vitórias .*

No primeiro exemplo, estamos perante sequências de palavras de duas orações distintas, separadas por vírgula, que emparelharam com a RegExp do provérbio **MP_290** *Querer é poder*, pois nela admitia-se a inserção de pontuação ('#'): `querer# <ser,V> <poder,V>.` No segundo exemplo, mais curioso, o nome *contrapoder*, cujo lema na STRING é *poder*, também foi capturado pela expressão regular. No exemplo seguinte, a expressão *Salve-se quem puder* está integrada como discurso reportado (entre aspas) e como sujeito de *poderá*, pelo que emparelha com a RegExp: `quem <poder,V># <poder,V>.` do provérbio **MP_271** *Quem pode, pode*. No segundo caso, estamos perante uma construção de verbo auxiliar modal *procurar* + infinitivo que se confunde com esta variante do provérbio **MP_272** *Quem procura acha*.

Em termos de diversidade de UP distintas presentes em cada *corpus*, observa-se valores semelhantes: o CTP apresenta ocorrências de 241 UP diferentes, o DSP 230 e o PRL apenas 200. Considerando como *densidade* (*Dens*) o n.º de palavras de cada *corpus* a dividir pelo nº de provérbios nele encontrados, verifica-se que Dens(CTP)=78.562, Dens(DSP)=30.279 e Dens(PRL)=68.915, ou seja, o *corpus* CTP apresenta uma maior densidade destas expressões (2,6 vezes relativamente ao DSP e 1,1 vezes mais em relação ao PRL). A densidade média dos provérbios do MP no conjunto dos 3 *corpora* é de 1 provérbio por cada 54.424 palavras. Quando comparamos as séries de valores de frequência das unidades paremiológicas encontradas em cada *corpus*, verifica-se uma correlação média-alta entre os dados obtidos nos *corpora* CTP e DSP (Pearson=0,691) e entre o CTP e o PRL (Pearson=0,671) mas uma correlação média-baixa entre o DSP e o PRL (Pearson=0,379). Neste sentido, a distribuição dos provérbios nos *corpora* DSP e PRL é mais dissemelhante do que a que se verifica quando cada *corpus* é comparado com o o CTP. Finalmente, verifica-se que metade (159; 50%) dos provérbios do MP ocorrem nos 3 *corpora*; cerca de um quarto (77; 24%) ocorre apenas em 2 dos *corpora* (a maioria no CTP e no DSP: 48; 15%); 40 provérbios (13%) ocorrem apenas num dos *corpora*, distribuindo-se de forma equilibrada; e apenas 42 não ocorreram em nenhum dos 3 *corpora*. A Fig. 1 apresenta (por ordem de ID) a distribuição dos 10 provérbios mais frequentes do MP pelos 3 *corpora* e que representam 25% do total de provérbios encontrados. Ora, apesar de, neste conjunto dos provérbios mais frequentes, a maioria (50%) das ocorrências se encontrar no DSP, verifica-se claras assimetrias da distribuição de cada provérbio.

Estes provérbios parecem também ser mais neutros do ponto de vista dos tópicos a que se podem aplicar e respetivos contextos de uso em que podem ocorrer (exceto, talvez, o **MP_185** *O segredo é a alma do negócio*). Veja-se, por exemplo, alguns dos contextos em que ocorre o provérbio **MP_133** *Mais vale tarde do que nunca*, que é, quanto ao total de ocorrências, o mais frequente nestes 3 *corpora*:

[CTP] *Depois de um complexo processo burocrático , a inauguração de uma embarcação para o transporte de 22 passageiros em o passado mês de Agosto acabaria por ser saudada com um*

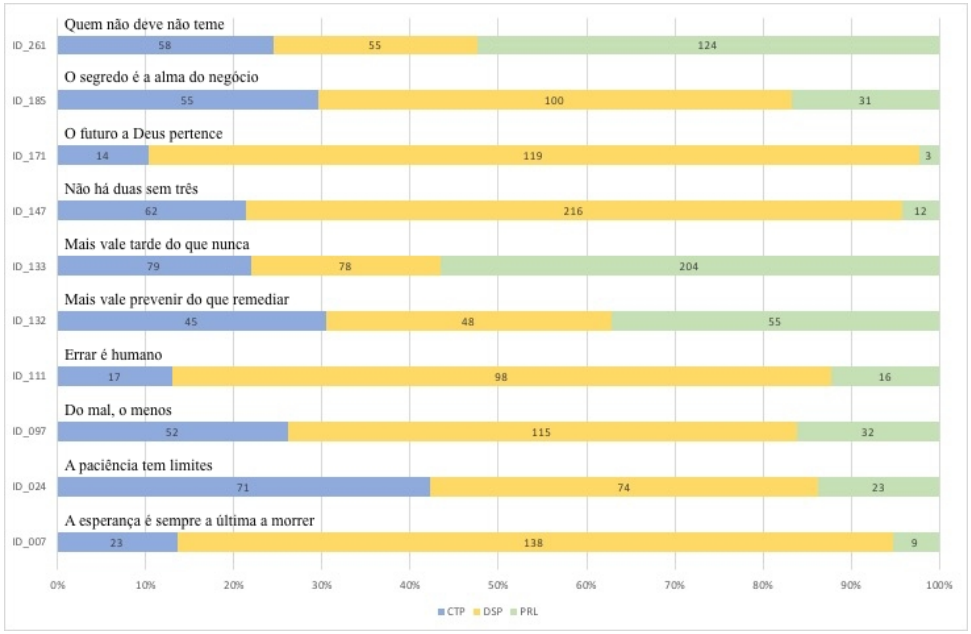


Figura 1. Distribuição dos 10 provérbios mais frequentes do MP pelos 3 *corpora*.

"**mais vale tarde do que nunca**".
[DSP] Antes tarde que nunca *: em a segunda parte , Scolari colocou em campo os dois Juninhos (Pernambucano e Paulista) e com isso ganhou maior eficácia em o meio-campo .*
[PRL] *Sr. Presidente , Sr. Primeiro-Ministro , em primeiro lugar , relativamente a as duas medidas que anunciou , era importante lembrar que* "**mais vale tarde do que nunca**" .

A aparente neutralidade temática deste provérbio deveria permitir que o mesmo ocorresse em diferentes situações comunicativas. Contudo, relativamente à sua distribuição nos 3 *corpora*, verifica-se que este provérbio ocorre praticamente o mesmo número de vezes no CETEMPúblico e no Desportivo (79 e 78 vezes), mas aparece 204 vezes no Parlamento. Considerando *frequência média* (*Frqmed*), como o n.º ocorrências do provérbio dividido pelo n.º total de ocorrências de todos os provérbios num dado *corpus*, temos para este provérbio: Frqmed(CPT)=3,53%, (DSP)=2,35% e (PRL)=11,37%, ou seja, o provérbio é 3 a 5 vezes mais frequente no PRL do que nos outros dois *corpora*, no CTP e no DSP, respetivamente. Como explicar esta assimetria? Veja-se, também, o que se passa com os provérbios com maior assimetria neste conjunto: **MP_171** *O futuro a Deus pertence*, em que 119 (88%) das ocorrências se encontram no *corpus* DSP, e **MP_007** *A esperança é sempre a última a morrer* com 170 (81%) instâncias no mesmo *corpus* DSP. Ambos os provérbios têm uma frequência baixa no *corpus* CTP (14 e 23 ocorrências) e praticamente residual no PRL (3 e 9 ocorrências), respetivamente. Dificilmente é possível detetar alguma razão no conteúdo e nos contextos de uso destes provérbios que justifique esta distribuição.

Dos 318 provérbios do MP, há 42 (13,2%) que não aparecem em nenhum dos 3 *corpora*. Trata-se, nalguns casos, de provérbios tematicamente bastante mais marcados do que os da lista acima, o que poderá explicar a sua ausência neste tipo de textos: um ideal de mulher: **MP_020** *A mulher e a sardinha querem-se da mais pequenina*; a alimentação: **MP_016** *A laranja, de manhã é ouro, à tarde é prata e à noite mata*; os filhos: **MP_114** *Filhos criados, trabalhos dobrados*; etc. Noutros casos, são provérbios que ditam uma filosofia de vida, de ordem geral, e.g. **MP_065**

Cada qual com o seu igual; **MP_159** *No meio é que está a virtude*; **MP_204** *Os opostos atraem-se*; **MP_247** *Quem espera sempre alcança*; **MP_275** *Quem sai aos seus não degenera*; **MP_316** *Vivendo e aprendendo*, sendo mais difícil explicar por que razão não ocorreram numa quantidade tão apreciável de textos. A maioria (237, 74%) dos provérbios do MP aparece com uma frequência inferior a 50 ocorrências. Alguns destes são também tematicamente bastante marcados, como, por exemplo, **MP_011** *A fome é o melhor tempero*, **MP_035** *Abril, águas mil*, ou **MP_117** *Gordura é formosura*. Outros, nem por isso: **MP_032** *A vida são dois dias*, **MP_042** *Amigo não empata amigo*, **MP_088** *Deitar cedo e cedo erguer, dá saúde e faz crescer*. Destes, 24 provérbios ocorrem apenas uma vez: **MP_148** *Não há regra sem exceção*, **MP_175** *O primeiro milho é dos pardais*, **MP_177** *O que arde cura*.

5. Conclusão

Neste artigo, apresentámos sucintamente os desafios que pela sua natureza, variação formal e modo de funcionamento nos textos, os provérbios levantam ao seu processamento computacional. De seguida, delineámos a metodologia seguida na constituição do Mínimo Paremiológico do Português Europeu (MP), constituído pelos 318 provérbios mais usuais, de acordo com critérios de frequência em *corpora* de natureza variada e de disponibilidade lexical. Os provérbios estão organizados em Unidades Paremiológicas (UP), que agregam a um mesmo provérbio o conjunto de variantes que este permite, recolhidas de diversas fontes. Em seguida, descrevemos o modo como este léxico de provérbios e suas variantes foram formalizados em gramáticas locais para integrar a STRING, de modo a permitir a sua identificação em textos. Finalmente, usando os resultados do processamento, descrevemos a distribuição dos provérbios em 3 *corpora* de natureza distinta e de grande dimensão.

A aplicação das gramáticas locais aos 3 *corpora* permitiu a identificação com elevada precisão (99,8%) de mais de 7.300 expressões proverbiais, nas suas múltiplas variantes. O método apresentado é, pois, válido e pode ser aplicado a textos de diversa natureza. Confirma-se, talvez dada a natureza destes *corpora*, a baixa frequência dos provérbios em textos escritos. Os poucos casos de falsos-positivos encontrados (#10) resultam sobretudo ou de inserções de elementos lexicais, ou de sinais de pontuação espúrios, ou ainda de problemas gerais de ambiguidade. O estudo da distribuição dos provérbios pelos três *corpora* revela a diversidade de situações e contextos de usos, dada a diversidade de unidades paremiológicas encontradas em cada *corpus*. Contudo, não é possível discernir padrões consistentes que expliquem, dada a natureza dos textos, as assimetrias encontradas na distribuição de cada unidade paremiológica pelos 3 *corpora* utilizados, seja quanto à sua frequência média, seja quanto às temáticas a que se aplicam. Este resultado é, em parte, esperável, já que, tratando-se dos provérbios mais usuais, seria esperável que apresentassem uma distribuição mais lata. Este artigo permitiu, assim, a construção de um conjunto de textos com provérbios usuais anotados, que ficará disponível para a comunidade científica. No futuro, a exploração de outro tipo de fontes textuais, ainda que suscite certas cautelas metodológicas na sua validação, deverá permitir encontrar outros contextos e novas variantes, validando eventualmente a intuição de que se trata de um tipo de unidade linguística mais ligada a um registo informal e a contextos predominantemente orais. Outra dimensão a explorar é a dinâmica temporal dos provérbios, sobretudo quando é possível relacionar estes com os tópicos (ou mesmo os acontecimentos) que motivaram o seu emprego, como sucede nos *corpora* do Desportivo e do Parlamento. Finalmente, pretende-se integrar a informação sobre os provérbios nos *corpora* já processados, permitindo a pesquisa *on-line* no demonstrador da STRING [Trindade 2020][3].

Agradecimentos. Parte da investigação para este artigo foi financiada por fundos públicos, pela Fundação para a Ciência e a Tecnologia (UIDB/50021/2020).

[3] `http://string.hlt.inesc-id.pt`

Referências

Costa, J. (1999). *O Livro dos Provérbios Portugueses*. Editorial Presença, Lisboa.

Davis, E., Danforth, C. M., Mieder, W., and Dodds, P. S. (2021). Computational paremiology: Charting the temporal, ecological dynamics of proverb use in books, news articles, and tweets. http://arxiv.org/abs/2107.04929.

Machado, J. (2011). *O Grande Livro dos Provérbios*. Casa das Letras, (4ª ed.), Alfragide.

Mamede, N., Baptista, J., Diniz, C., and Cabarrão, V. (2012). STRING - A Hybrid Statistical and Rule-Based Natural Language Processing Chain for Portuguese. In Abad, A., editor, *International Conference on Computational Processing of Portuguese (PROPOR 2012) - Demo Session*, Coimbra, Portugal. http://www.propor2012.org/demos/DemoSTRING.pdf.

Marques, J. (2013). Anaphora resolution. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, L^2F/INESC-ID, Lisboa.

Mendes, R. and Oliveira, H. G. (2020a). Comparing different methods for assigning Portuguese proverbs to news headlines. In Mikolov, T., Yih, W.-T., and Zweig, G., editors, *Linguistic regularities in continuous space word representations. Proceedings of NAACL-HLT, NAACL.*, pages 746–751. 11th International Conference on Computational Creativity (ICCC'20), ACL.

Mendes, R. and Oliveira, H. H. (2020b). TeCo: Exploring Word Embeddings for Text Adaptation to a given Context. In *Proceedings of ICCC*. 11th International Conference on Computational Creativity (ICCC'20), ACL.

Mitkov, R. (2002). *Anaphora Resolution*. Pearson – Prentice Hall.

Moreira, A. (1996). *Provérbios Portugueses*. Editorial Notícias, Lisboa.

Parente, S. (2005). *O Livro dos Provérbios*. Editora Âncora, Lisboa.

Rassi, A. P., Baptista, J., and Vale, O. A. (2014a). Proverb variation: Experiments on automatic detection in Brazilian Portuguese texts. In Baptista, J., Mamede, N., Candeias, S., Paraboni, I., Pardo, T., and Volpe Nunes, M., editors, *Computational Processing of the Portuguese Language*, volume 8775 of *Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence*, pages 141–152, Berlin. 11th International Conference PROPOR'2014, São Carlos – SP, Brazil, October 8-10, 2014, Springer.

Rassi, A. P., Vale, O. A., and Baptista., J. (2014b). Automatic detection of proverbs and their variants. In Pereira, M., Leal, J., and Simões, A., editors, *Proceedings of the Symposium on Languages, Applications and Technologies (SLATE'14)*, pages 235–250, Leibniz (Germany). Symposium on Languages, Applications and Technologies (SLATE'14), Bragança (Portugal), June 19-20, 2014., Schloss Dagstuhl - Leibniz-Zentrum fur Informatik, Dagstuhl Publishing.

Reis, S. (2020). *Expressões proverbiais do português: Usos, variação formal e Identificação automática.* PhD thesis, Universidade do Algarve, Faro, Algarve, Portugal.

Reis, S. and Baptista, J. (2016a). Estimating lexical availability of european portuguese proverbs. In Mitkov, R. and Corpas Pastor, G., editors, *EUROPHRAS 2017*, volume 10596 of *Lecture Notes in Computer Science*, pages 232–244, Cham. Springer.

Reis, S. and Baptista, J. (2016b). Let's Play with Proverbs? NLP Tools and Resources for iCALL Applications around Proverbs for PFL. In *Proceedings of the International Interdisciplinary Conference in Social and Human Sciences*, Faro, Portugal. University of Algarve, Faculty of Economics.

Reis, S. and Baptista, J. (2016c). O uso de provérbios no ensino de português. In Soares, R. & Lauhakangas, O. (Eds.) *10th Interdisciplinary Colloquium on Proverbs*, Actas ICP16 Proceedings. Tavira: AIP-IAP, 2017, pp. 521–538.

Reis, S. and Baptista, J. (2017). Os provérbios em manuais de ensino de português língua não materna. In Vládia Pinheiro & Gustavo Henrique Paetzold (Eds.) *Proceedings of Symposium in Information and Human Language Technology* Uberlandia, MG, Brazil, October 2-5, 2017, Sociedade Brasileira de Computação, pp. 247–255.

Reis, S. and Baptista, J. (2020). Determinação de um mínimo paremiológico do português europeu. *Acta Scientiarum. Language and Culture*, 42(2):e52114. https://doi.org/10.4025/actascilangcult.v42i2.52114.

Santos, D. and Rocha, P. (2001). Evaluating CETEMPúblico: A Free Resource for Portuguese. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 442–449, Toulouse, France.

Talhadas, R., Baptista, J., and Mamede, N. (2013). Semantic roles annotation guidelines. Technical report, L2F/INESC ID Lisboa.

Trindade, J. (2020). Syntax Deep Explorer: Integrating multi-corpora support into a corpus analysis tool. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, L2F/INESC-ID, Lisboa.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Descrição Preliminar do *Corpus* DANTEStocks: Diretrizes de Segmentação para Anotação segundo *Universal Dependencies*

Ariani Di Felippo[1], Caroline Postali[1], Gabriel Ceregatto[1], Laura S. Gazana[1], Emanuel H. da Silva[2], Norton T. Roman[3], Thiago A. S. Pardo[2]

[1]Núcleo Interinstitucional de Linguística Computacional (NILC)
Departamento de Letras – Universidade Federal de São Carlos (UFSCar)
Caixa Postal 676 – 13565-905 – São Carlos – SP – Brasil

[2]Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (USP)
Caixa Postal 668 – 13566-970 – São Carlos – SP – Brazil

[3]Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)

ariani@ufscar.br, caroline.postali@gmail.com,
gabriel@ceregatto.admin.br, lauragazana@estudante.ufscar.br,
{emanuel.huber,norton}@usp.br, taspardo@icmc.usp.br

Abstract. *The annotation of informal texts within the Universal Dependencies framework requires two segmentation processes: definition of the relevant unity for syntactic analysis and identification of syntactic words. In this paper, we present the linguistic idiosyncrasies of DANTEStocks, a corpus of tweets from the financial market, written in Portuguese, and the general guidelines for their automatic segmentation. As such, this work contributes to a better understanding of linguistic aspects of tweets and the development of resources and tools for automatic processing of this subgenre of user-generated content.*

Resumo. *A anotação de textos informais segundo a Universal Dependencies requer dois processos de segmentação: delimitação da unidade relevante para a análise sintática e identificação das palavras sintáticas. Neste artigo, apresentam-se as idiossincrasias linguísticas do corpus DANTEStocks, composto por tweets do mercado financeiro, escritos em Português, e as estratégias gerais de segmentação automática. Assim, contribui-se para a descrição de aspectos linguísticos dos tweets e para o desenvolvimento de recursos e ferramentas de processamento automático desse subgênero de "user-generated content".*

1. Introdução

Diante da imensa relevância adquirida na última década, as redes sociais (como *Facebook*, *WhatsApp*, *Twitter*, etc.) são fontes de conteúdo (em inglês, *user-generated content* - UGC) inestimáveis para consumidores, políticos e governos no geral. Com isso, o desenvolvimento de ferramentas e aplicações linguístico-computacionais (como as de análise de sentimento e mineração de opinião) tem se tornado tópico central do Processamento Automático das Línguas Naturais (PLN) [Sanguinetti et al., 2020a].

Nesse cenário, já há vários *taggers* (etiquetadores morfossintáticos) [p.ex.: Owoputi et al., 2013; Lynn et al., 2015; Bosco et al., 2016; Proisl, 2018] e *parsers* (analisadores sintáticos) [p.ex.: Foster, 2010; Petrov, Mcdonald, 2012; Kong et al., 2014 e Liu et al., 2018] relativamente precisos para o processamento de UGCs, sobretudo em inglês. E esse ferramental só foi desenvolvido graças aos *corpora* anotados (*treebanks*) e aos algoritmos de aprendizado de máquina. Grande parte dos *treebanks* de UGC construídos nos últimos anos são compostos exclusivamente por *tweets*. O destaque dos *corpora* de *tweets* (os *tweebanks*) se deve pela facilidade de obtenção dos dados, política do *Twitter* sobre o uso dos dados para fins acadêmicos e relevância para aplicações de PLN. O tamanho desses recursos varia de 500 a aproximadamente 6,700 mensagens [Sanguinetti et al., 2020a].

Os *tweebanks* mais recentes possuem anotação segundo a *Universal Dependencies* (UD) [Nivre, 2015; Nivre et al., 2020], um modelo gramatical que fornece principalmente um conjunto de etiquetas morfossintáticas universais e de relações de dependências sintáticas para anotação de *corpus,* o que possibilita estudos "*cross*-linguísticos" e reuso de metodologias.

A anotação UD de UGC, como os *tweets,* requer inicialmente que a unidade relevante para a análise sintática seja definida. Isso significa decidir se essa unidade será delimitada com base na noção de sentença (como nos textos formais) ou outro critério. Ademais, por se basear em uma visão lexicalista da sintaxe, a anotação UD necessita que as palavras sintáticas[1] sejam identificadas (tokenizadas)[2]. Para tanto, é preciso descrever as características linguísticas (estruturais, ortográficas e lexicais) do *tweets* que compõem o *corpus* que será anotado [Liu et al., 2018; Sanguinetti et al., 2020a,b]. Por exemplo, uma característica geral dos *tweets* é a ocorrência de autocensuras ("m*" → "merda"). Para o reconhecimento das autocensuras como palavras, é preciso prever que o asterisco não seja segmentado, mas reconhecido como parte constitutiva do *token.*

Neste artigo, descrevem-se as características linguísticas do *corpus* construído por Silva et al. (2020) e as decorrentes estratégias automáticas de segmentação (isto é, delimitação da unidade de análise sintática e tokenização) para anotação UD. Denominado DANTEStocks, o *corpus* é composto por *tweets* em português sobre ações do índice Ibovespa e possui anotação de emoções. O DANTEStocks será o primeiro *corpus* de UGC em português com anotação UD. Acredita-se que a anotação UD poderá potencializar o emprego do *corpus* nas investigações sobre análise de sentimentos e ampliar a sua utilidade em outros tipos de pesquisas linguístico-computacionais. Dessa forma, esse trabalho contribui para os estudos descritivos sobre as características linguísticas dos *tweets* e para o desenvolvimento de recursos, ferramentas e aplicações de processamento automático desse tipo particular de UGC.

Nas Seções 2, descreve-se brevemente o modelo UD. Na Seção 3, apresentam-se o *corpus* DANTEStocks, as características estruturais de seus *tweets* e a decorrente delimitação da unidade de análise sintática. Na Seção 4, sistematizam-se os dispositivos linguísticos (lexicais e ortográficos) que caracterizam o *corpus* e discute-se a tokenização de alguns deles. Na Seção 5, apresentam-se as considerações finais sobre o trabalho, destacando suas contribuições e estudos futuros.

[1] Palavra sintática (em inglês, *syntactic word*) é a unidade mínima a que corresponde uma função sintática (https://universaldependencies.org/u/overview/tokenization.html).
[2] Na anotação UD, palavras sintáticas (ou itens lexicais) são sinônimos de *tokens*.

2. O Modelo *Universal Dependencies*

O modelo UD prevê anotação no nível sentencial e diretrizes para tokenização e anotação morfossintática e sintática[3]. Sobre a tokenização, a UD, a partir de uma visão lexicalista da sintaxe, define que uma relação de dependência (*deprel*) ocorre entre palavras de uma sentença e as características morfológicas são representadas por propriedades (ou *features*). Assim, as unidades básicas de anotação são palavras sintáticas. Com isso, os clíticos precisam ser separados de seus hospedeiros ("prepare-se" → "prepare" "se") e tratados como palavras independentes, assim como as contrações precisam ser decompostas ("das" → "de" "as"). Excepcionalmente, o modelo permite a combinação de *tokens* ortográficos em uma única palavra, como é o caso das abreviações (p.ex.: "e.g."). Quanto à anotação linguística, a UD prevê 2 níveis. No nível morfológico, especificam-se 3 tipos de informação: lema, etiqueta morfossintática e traços lexicais/gramaticais (das palavras). No nível sintático, parte-se da premissa de que as deprels são relações binárias e assimétricas [Nivre, 2015; Nivre et al., 2020; Marnefee et al., 2021] e que a representação básica de uma estrutura de dependências é arbórea, na qual uma palavra é o *root* (raiz) da sentença.

Na Figura 1, ilustra-se a anotação UD de uma sentença de um *corpus* jornalístico em português. Em caixa alta, estão codificadas as etiquetas morfossintáticas, como DET para "esse", NOUN para "carro" e VERB para "achado". A versão 2.0[4] da UD dispõe de 17 etiquetas, juntamente com critérios para o emprego de cada uma delas. Logo acima das etiquetas, estão as formas canônicas, por exemplo: "esse", "carro" e "achar" são respectivamente os lemas de "esse", "carro" e "achado". As *deprels* estão indicadas por setas rotuladas que se originam no *head* (cabeça) e se destinam ao dependente. Na Figura 1, "carro", por exemplo, é dependente de "achado" (cabeça) e estes estão conectados pela *deprel* **nsubj:pass** (sujeito nominal da passiva). O verbo "achado" é a raiz da sentença-exemplo. A UD (2.0) fornece 37 relações, juntamente com critérios para o emprego de cada uma delas. A UD também fornece uma lista bastante extensa de traços que codificam propriedades lexicais e gramaticais das palavras. Embora ausentes na Figura 1, "carro", no caso, possui os traços-valores: Gender=Masc e Number=Sing[5].

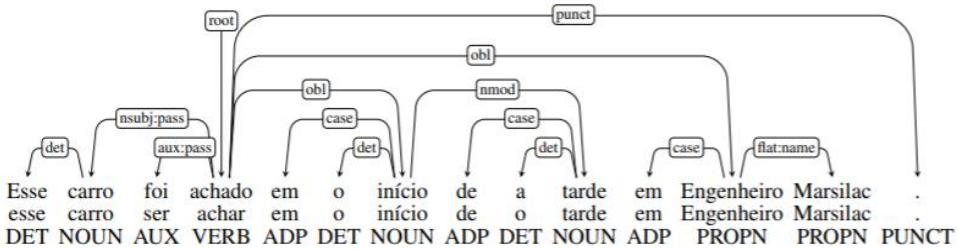


Figura 1. Exemplo de anotação sintática segundo a UD [Rademaker *et al*., 2017].

A seguir, apresenta-se o DANTEStockes para, na próxima seção, descrever as suas particularidades lexicais, ortográficas e estruturais.

[3] Para o português do Brasil, Duran (2021) construiu um manual com diretrizes de tokenização e de anotação morfossintática segundo a UD (especificamente para textos formais, como os jornalísticos).
[4] https://universaldependencies.org/guidelines.html
[5] Essa informação foi recuperada do *corpus UD-Portuguese-Bosque* por meio da plataforma *online* Grew-match (http://match.grew.fr/?corpus=UD_Portuguese-Bosque@2.8).

3. DANTEStocks - Estrutura dos *Tweets* e Definição da Unidade de Análise

O DANTEStocks[6] é um *corpus* de material textual compilado do *Twitter*, que parece uma mescla de rede social e *microblog*[7] [Freitas, Barth, 2015], e cujas principais características são a dinamicidade das interações (sejam comentários ou republicações) e a brevidade das mensagens (restrição de 140 caracteres). Considerado um gênero, o *tweet* parece ser constituído por resquícios de outros gêneros (como notícia, propaganda, bilhete, diário íntimo, etc.), que foram modificados para atender às necessidades de comunicação da rede [Marcuschi, 2008, Freitas, Barth, 2015]. Aliás, esses diferentes gêneros que se entrelaçam nos *tweets* evidenciam a influência da oralidade nessa escrita online. O DANTEStocks engloba especificamente 4.517 *tweets* contendo menção a alguma das ações do índice Ibovespa[8]. As postagens foram coletadas automaticamente em 2014 com base nos *tickers* (códigos) (isto é, cadeias de 4 letras e 1 número que fazem alusão ao nome da empresa e ao tipo de ação, como "PETR4") das ações do índice. O *corpus*, originalmente construído para pesquisas sobre análise de sentimentos, já possui anotação de emoções [cf. Silva et al., 2020].

Quanto à composição estrutural, os *tweets* do *corpus* variam bastante. Há *tweets* formados por uma ou mais sentenças claramente delimitadas, como (1), (2) e (3). Mas há também *tweets* que apresentam, frente às normas da língua padrão, ausência de pontuação (4) ou pontuação equivocada (5). *Tweets* relativamente fragmentados (6) também compõem o *corpus*. Em (4), o *tweet* parece ser composto por duas sentenças ("O #PT conseguiu fazer propaganda eleitoral antecipada" e "O que a @dilmabr tem a dizer sobre isso"). Essa interpretação pode ser corroborada pela capitalização do segundo "o" (negrito). Em (5), o exemplo é de uso inadequado da vírgula, provavelmente em substituição ao ponto de exclamação. O *tweet* (6) exemplifica uma postagem relativamente fragmentada, composta por uma *hashtag* seguida por um sintagma nominal e um *link.* O *tweet* (3), em especial, apresenta alternância de código linguístico (em inglês, *code-switching*) (português-inglês) em nível sentencial.

(1) Sera k petr4 já entrou na baixa?
(2) PETR4 subiu na bolsa 13,50. Muito bem, surpreso com o resultado.
(3) #CSNA3: Está em região de suporte que vem resistindo. Who knows?
(4) O #PT conseguiu fazer propaganda eleitoral antecipada **O** que a @dilmabr tem a dizer sobre isso?
(5) Bom dia Marcos, Alguma previsão para petr4?!
(6) #GGBR4 Suportes e resistências http://t.co/Azw6yIEVI9

Para a anotação sintática de textos formais, a unidade de anotação é comumente a sentença, cuja segmentação automática é tarefa relativamente simples, pois a pontuação pode ser usada como critério [Reynar e Ratnaparkhi, 1997]. Buscando estabelecer certa compatibilidade com os *treebanks* de textos formais, Sanguinetti et al. (2020b) optam por segmentar (automaticamente) somente os *tweets* com sentenças bem delimitadas (como (1), (2) e (3)), podendo utilizar índices para reconstruir, se necessário, as mensagens segmentadas [Rehbein et al., 2019].

[6] Disponível em: https://www.kaggle.com/fernandojvdasilva/stock-tweets-ptbr-emotions/data.
[7] *Microblog* é um tipo de *blog* no qual os usuários fazem atualizações breves de texto (até 200 caracteres), sobretudo veiculando impressões pessoais.
[8] Principal índice da bolsa de valores oficial do Brasil, a B3 (de "Brasil, Bolsa, Balcão").

Em outros trabalhos, o *tweet* é considerado unidade mínima de anotação [Kong et al., 2014; Liu et al., 2018; Sanguinetti et al., 2018]. Embora a não-segmentação dos *tweets* em unidades menores possa levar a um emprego excessivo da parataxis[9] [Sanguinetti et al., 2020], que é a *deprel* usada para relacionar elementos justapostos que não estejam coordenados, subordinados ou em outra relação argumento-predicado, essa foi a opção adotada para o DANTEStocks. Com isso, a anotação UD (morfossintática, por enquanto) do *corpus* está sendo feita no nível do *tweet*.

Essa opção se justifica por algumas razões. Uma delas são os problemas de pontuação, que dificultam a segmentação sentencial automática. Embora haja outros critérios para essa segmentação, como a detecção de estruturas verbo-argumento, estes não foram considerados devido à complexidade de se processar automaticamente os *tweets.* Assim, considerar o *tweet* como unidade única economiza o esforço necessário para desenvolver, manter, adaptar ou realizar o pós-processamento em um segmentador automático. Além disso, considerar o *tweet* como unidade mínima pode ser relevante para pesquisas linguístico-computacionais a respeito desse tipo de UGC. Cignarella *et al*. (2019), por exemplo, destacam que o estudo da correlação entre aspectos sintáticos (via UD) e ironia só foi possível diante dos *tweets* enquanto unidade. Outra razão importante é a interpretação (e consequente anotação sintática) dos *tweets*, que, muitas vezes, depende da mensagem completa. Isso fica evidente diante dos *tweets* que têm certa fragmentação, como (6). Somente é possível interpretar que os níveis de "suporte" e "resistência" (isto é, conceitos de análise gráfica) de interesse são os relativos à ação/*ticker* "GGBR4" com base na mensagem completa. A anotação intersentencial de alternância de código também pode ser considerada mais apropriada no nível do *tweet*.

4. Os Fenômenos UGC do DANTEStocks e a sua Tokenização

A partir de trabalhos como os de Lyddy et al. (2014), Liu et al. (2018) e Sanguinetti et al., 2020a,b), as particularidades ortográficas e lexicais do *corpus* foram sistematizadas em 7 dimensões. As dimensões e os fenômenos estão exemplificados no Quadro 1.

1. **Simplificação de código:** engloba os fenômenos "ergográficos" (em inglês, *ergographic phenomena*), que reduzem o esforço de escrita de um único *token*, como remoção/adição de diacrítico, ausência de hífen, substituição de diacrítico (pela letra "h"), omissão de letras (finais e mediais), erro ortográfico/digitação e fonetização.
2. **Abreviação:** toda sequência de caracteres que representa de forma reduzida várias palavras; a abreviação pode ser do tipo contração (de elementos gramaticais), acrônimo ou inicialismo (do inglês, *initialism*) (isto é, abreviações compostas pelas letras iniciais de palavras comuns ("lp" → "longo prazo") [Lyddy et al., 2014].
3. **Expressão de sentimento:** fenômenos que emulam o sentimento expresso pela prosódia, expressão facial ou gesto na interação via *tweet,* como alongamento grafêmico (sobretudo de vogais), repetição de pontuação, autocensura e emoticons.
4. **Influência de língua estrangeira:** vocábulo formado com base em outra língua; "estopar", por exemplo, baseia-se no verbo em inglês *"stop"* ("parar") (isto é, interromper venda ou compra de um ativo diante de dado preço).
5. **Expressão de oralidade:** toda palavra cuja grafia remonta à comunicação (fala) informal, as quais são, por vezes, empregadas com função humorística.

[9] https://universaldependencies.org/u/dep/parataxis.html

6. **Elemento metalinguístico**: todo elemento que tipicamente ocorre no *Twitter*, como *hashtag*, menção, marca de *retweet*, URL e truncamento lexical (quebra de palavra).
7. **Fenômeno de domínio:** todo fenômeno lexical/gráfico que diferencia os *tweets* do DANTEStocks dos demais *tweets*, a saber: *tickers*, *cashtag*, numerais com parte decimal indeterminada, índices de (des)valorização das ações, substituições lexicais (por símbolo), expressões temporais alfanuméricas e valor monetário aglutinado.

Quadro 1. Exemplo dos fenômenos UGC no DANTEStocks.

Fenômeno	**Exemplo**	**Forma padrão/glosa[10]**
Simplificação de código		
Ausência/adição de diacrítico	**proprio, milhao, Graca, fêz**	*próprio, bilhão, Graça, fez*
Ausência de hífen	**sexta feira, caça níquel**	*sexta-feira, caça-níquel*
Substituição de diacrítico	**eh, neh, tou**	*é, né, tô*
Omissão de letras	**d, n, qdo, tx, ult, pq**	*de, não, quando, taxa, último, porque*
Erro ortográfico/digitação	**comrpa, agradeveis**	*compra, agradáveis*
Fonetização	**k, kd**, **krk, kct**	*que, cadê, caraca, cacete*
Abreviação		
Contração	**oq, pq**	*o que, por que, por favor*
Acrônimo/inicialismo	**BB, cf, lp**	*Banco do Brasil, conselho fiscal, longo prazo*
Expressão de sentimento		
Alongamento de pontuação	Onde a #OIBR4 vai parar**???**	Onde a #OIBR4 vai parar**?**
Alongamento grafêmico	n**oo**s**aaa**, LI**NN**DA	*nossa, linda*
Autocensura	**p**a m***	*puta, merda*
Emoticon	**o.O :) :/**	*surpresa, sorriso (feliz), indecisão*
Influência de língua estrangeira		
Formação verbal	**estopar**	*'parar investimento'*
Marca de oralidade		
Coloquialismo	**guvêrno, bão, ae, péra, vamu**	*governo, bom, aí, espere, vamos*
Expressão cristalizada	**né, daí (dae)**	*'não é', 'de aí'*
Exclamação onomatopeica	**hahaha, hehehe**	*risos*
Elementos metalinguísticos (do *Twitter*)		
Hashtag	**#Petr4**	*'indexadores de tópicos ou assuntos'*
Menção	**@garimpodeacoes**	*'perfil/usuário'*
Marca de *retweet*	**RT** @Ary_AntiPT	*'republicação de um tweet'*
URL	**http://t.co/sROpyWPblN**	*'endereço da web'*
Truncamento (lexical)	Ação sobe **fo...**	*Ação sobe fo(rte)*
Fenômeno do domínio (Ibovespa)		
Ticker	**Petr4**	*'código de uma ação'*
Cashtag	**$LREN3**	*'código de ação precedido por $'*
Indeterminação da parte decimal	De 18,**xx** a 21,00	*'qualquer valor na parte decimal'*
Índice de (des)valorização	**+2,09%**, **-11,42%**	*'percentual de (des)valorização de ação'*
Substituição lexical	... precisam de muito **$**	*... precisam de muito dinheiro*
Expressão (temporal) híbrida	**1T14**	*primeiro trimestre de 2014*
Valor monetário aglutinado	**R$20,00**	*R$ 20,00*

[10] As formas de superfície do *corpus* não foram substituídas pelas formas da linguagem padrão, as quais estão no Quadro 1 apenas como recurso didático fornecido ao leitor para a compreensão dos fenômenos.

Partindo-se da decisão de não normalizar os *tweets* do *corpus* com o objetivo de desenvolver ferramentas e sistemas para o mundo real, foi necessário definir o estatuto de palavra de alguns dos fenômenos sistematizados para a subsequente tokenização.

Quanto aos fenômenos de simplificação de código, ressalta-se que um composto hifenizado (como "caça-níquel") constitui, segundo a visão lexicalista da UD, uma única palavra. Assim, mesmo que a ausência do hífen, como em "caça níquel", resulte na identificação automática de dois *tokens* ("caça" e "níquel"), a anotação UD precisa evidenciar que se trata de um composto, isto é, *token* único. Uma alternativa pode ser a utilização da *deprel* **compound**, como é feito no *corpus* UD_English-EWT[11] em inglês.

As contrações são formas abreviadas de duas palavras funcionais com remoção de espaços e letras. Nessa categoria, no entanto, há diferentes fenômenos de redução, os quais necessitam, por isso, de estratégias distintas de tokenização. A forma superficial "oq" (em "Oq faz?"), por ser constituída por dois pronomes ("o" "que") e ter a função única de pronome, corresponde a um *token* único. Já "pq", ao reduzir duas palavras ("por" "que"), de categorias morfossintáticas diferentes (preposição e pronome, respectivamente), deve ser decomposta em dois *tokens*. Os outros tipos de abreviação, ou seja, acrônimos (que reduzem nomes de entidades), como "BB" ("Banco do Brasil"), e inicialismos (que abreviam expressões compostas por palavras comuns), como "cf" ("conselho fiscal"), são *tokens* únicos, uma vez que desempenham função sintática específica, sendo possível atribuir à forma reduzida a categoria morfossintática do *head*.

Quanto às expressões de sentimento, destaca-se que as autocensuras, como "car*" ("caralho"), e os *emoticons* (";-*" → "beijo") correspondem a palavras sintáticas. No entanto, o adequado reconhecimento destes como tal requer que os sinais de pontuação e os caracteres especiais sejam reconhecidos como elementos constitutivos do *token*. No DANTEStocks, os *emoticons* ocorrem ao final dos *tweets,* não havendo uma ligação clara com a estrutura do *tweet,* a não ser "discursiva".

As marcas de oralidade classificadas como "expressão cristalizada" – "né" e "daí" (ou "dae") – são etimologicamente contrações de "não é" e "de aí". Sendo contrações (ou seja, *tokens* compostos por mais de uma categoria gramatical), elas seriam tokenizadas segundo a UD. No entanto, essas expressões funcionam no *corpus* como uma unidade, sendo o mais adequado, nesse caso, não realizar a decomposição. Atualmente, a categoria gramatical mais adequada a ser atribuída a elas está sob estudo (se advérbio ou interjeição). No nível sintático, no entanto, sabe-se que essas expressões desempenham função discursiva e a anotação via *deprel* precisará evidenciar isso.

Sobre os elementos metalinguísticos, os truncamentos lexicais ocorrem principalmente no fim de um *tweet* devido ao limite de caracteres. Na literatura, eles são tokenizados e, caso as formas completas possam ser recuperadas, os truncamentos são anotados em função delas. No que diz respeito às *hashtags* (e também *cashtags*) e menções, o reconhecimento dos símbolos "$" e "@" como parte constitutiva dos *tokens* parece variar na literatura. No DANTEStocks, esses símbolos foram considerados como tal, compondo um *token* único com a palavra ou expressão que eles precedem.

Quanto aos fenômenos de domínio, os índices de (des)valorização das ações compreendem 3 *tokens* ("+2,09%" → "+" "2,09" "%"). Especificamente, reconhecer "+" como *token* (no caso, um símbolo) justifica-se pela possibilidade de substituí-lo por

[11] https://github.com/UniversalDependencies/UD_English-EWT

outra palavra (como "subiu"). Outra característica de domínio são as formas reduzidas de expressões temporais, como "1T14" ("primeiro trimestre de 2014"). Estas, ao funcionarem como unidade, são consideradas palavras únicas e anotadas com a categoria morfossintática do *head*, como sugerido para os acrônimos e inicialismos. No DANTEStocks, as expressões monetárias podem ocorrer aglutinadas (isto é, sem espaço entre o símbolo monetário e o numeral) ("R$20,00"). Estas, no entanto, são compostas por dois *tokens* (já que "R$20,00" é o mesmo que "vinte reais") e, por isso, precisam ser tokenizadas.

5. Considerações finais

A caracterização linguística ora apresentada revelou que os *tweets* do DANTEStocks são marcados por convenções e limitações impostas pela plataforma, marcas de informalidade e certos dispositivos linguísticos, alguns deles, aliás, dependentes de domínio. O estudo sobre a estrutura dos *tweets* e a descrição dos dispositivos lexicais e gráficos fundamentaram a segmentação do *corpus* para a anotação UD. A definição do estatuto de *token* dos fenômenos resultou em algumas regras contextuais utilizadas por Silva et al. (2021) para adaptar o tokenizador simbólico de *tweets* do pacote NLTK[12] ao DANTEStocks. Para dar continuidade a este trabalho, pretende-se quantificar os fenômenos no *corpus*, gerando estatísticas de frequência/relevância.

Agradecimentos

Os autores deste trabalho agradecem ao Centro de Inteligência Artificial (C4AI - USP) e o apoio da Fundação de Apoio à Pesquisa do Estado de São Paulo (processo FAPESP #2019/07665-4) e da IBM Corporation.

Referências

Bosco, C., Tamburini, F., Bolioli, A., Mazzei, A. (2016). Overview of the EVALITA 2016 Part of Speech tagging on TWitter for ITAlian task. In: Anas do 5º EVALITA.

Cignarella, A.T., Bosco, C., Rosso, P. (2019). Presenting TWITTIRO-UD: an Italian twitter treebank in Universal Dependencies. In: Anais do 5º Depling, p.190-7. Paris, França, ACL.

Duran, M.S. (2021). Manual de anotação de PoS tags. *Relatório Técnico*, n. 434. NILC-ICMC/USP, 54p. Disponível em: https://sites.google.com/icmc.usp.br/poetisa. Acesso em: 20/09/2021.

Eisenstein, J. (2013). What to do about bad language on the internet. In: Anais do NAACL-HLT, p. 359–369. Atlanta, EUA, ACL.

Foster, J. (2010). "cba to check the spelling": investigating parser performance on discussion forum posts. In: Anais do NAACL-HLT, p. 381–384. LA, EUA, ACL.

Freitas, E.C.; Barth, P.A. (2015) Gênero ou suporte? O entrelaçamento de gêneros no Twitter. *Revista (Con)Textos Linguísticos*, 9(12), p. 08-26.

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., Smith, N.A. (2014). A dependency parser for tweets. In: Anais do EMNLP, p. 1001–12. Doha, Qatar.

Lyddy, F., Farina, F., Hanney, J., Farrell, L., O'Neill, N.K. (2014). An analysis of language in university students' text messages. *Journal of Computer-Mediated Communication,* 19(3), p. 546-561. Wiley Online Library.

[12] https://www.nltk.org/api/nltk.tokenize.html

Lynn, T., Scannell, K., Maguire, E. (2015). Minority language Twitter: part-of-speech tagging and analysis of Irish tweets. In: Anais do ACL'15 Workshop on Noisy User-generated Text, p. 1–8. July 31. Beijing, China, ACL.

Liu, Y., Zhu, Y., Che, W., Qin, B., Schneider, N., Smith, N.A. (2018). Parsing tweets into Universal Dependencies. In: Anais do NAACL-HLT, p. 965–975. LA, EUA, ACL.

Marcuschi, L.A. Produção textual, análise de gêneros e compreensão. Parábola Ed., 2008.

De Marneffe, M-C., Manning, C.D., Nivre, J. Zeman, D. (2021). Universal Dependencies. In *Computational Linguistics*, 47(2), p. 255-308. ACL. Online ISSN 1530-9312.

Nivre, J. (2015). Towards a Universal Grammar for Natural Language Processing. In: Anais do CICLing 2015. Lecture Notes in Computer Science, vol 9041, p. 3-16, Ed. by A. Gelbukh. Springer, Cham.

Nivre, J. et al. (2020). Universal Dependencies v2: an evergrowing multilingual treebank collection. In: Anais do 12º LREC. P. 4034-4043. Marseille, França. ELRA.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., Smith, N.A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In: Anais do NAACL-HLT, p. 380–390. 9-14 de junho. Atlanta, Georgia. ACL.

Petrov, S., Das, D., McDonald, R. (2012). A universal part-of-speech tagset. In: Anais do 8º LREC, p. 2089–2096. 21-27 de maio. Istanbul, Turquia. ELRA.

Proisl, T. (2018). Someweta: A part-of-speech tagger for German social media and web texts. In: Anais do 11º LREC, p. 665–670. May 7-12. Miyazaki, Japão. ELRA.

Plutchik R., Kellerman, H. (eds). 1986. Emotion: theory, research and experience. Nova Iorque: Acad. Press

Rademaker, A.; Chalub, F., Real, L., Freitas, C., Bick, E., Paiva, V. (2017). Universal Dependencies for Portuguese. In: Anais do 4º Depling, p. 197-206. Pisa, Itália.

Rehbein, I., Ruppenhofer, J., Bich-Ngoc, D. (2019). tweeDe – a Universal Dependencies treebank for German tweets. In: Anais do 18º TLT, p. 100-108. Paris, França. ACL.

Reynar, J., Ratnaparkhi, A. (1997). A maximum entropy approach to identifying sentence boundaries. In: Anais do 5º ANLP, p. 16-19. Washington, EUA, ACL.

Seddah, D., Sagot, B., Candito, M., Mouilleron, V., Combet, V. (2012). The French social media bank: a treebank of noisy user generated content. In: Anais do 24º COLING, p. 2441–2458, Mumbai, Índia, ACL.

Sanguinetti, M. et al. (2018). PoSTWITA-UD: An Italian twitter treebank in Universal Dependencies. In: Anais do 11º LREC. p. 1768–75. Miyazaki, Japão. ELRA

Sanguinetti, M. et al. (2020a). Treebanking user-generated content: a proposal for a unified representation in universal dependencies. In: Anais do 12º LREC. p. 5240-50. Marseille, França. ELRA

Sanguinetti, M. et al. (2020b). Treebanking user-generated contente: a UD based overview of guidelines, corpora and unified recommendations. Available in: https://arxiv.org/abs/2011.02063. Access in: 25/09/2021.

Silva, F.J.V., Roman, N.T., Carvalho, A.M.B.R. (2020). Stock market tweets annotated with emotions. In: *Corpora*, 15(3), p. 343-354. Online ISSN: 1755-1676.

Silva, E.H., Pardo, T.A.S., Roman. N.T, Di-Felippo, A. Universal Dependencies for tweets in Brazilian Portuguese: tokenization and part of speech tagging. In: Anais do XVIII ENIAC 2021. 29 de nov. a 3 de dez., 2021. No prelo

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Descrição de numerais segundo modelo *Universal Dependencies* e sua anotação no português

Magali Sanches Duran, Lucelene Lopes, Thiago Alexandre Salgueiro Pardo

Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo
São Carlos, Brasil

magali.duran@gmail.com, lucelene@gmail.com, taspardo@icmc.usp.br

Abstract. *This paper describes the instantiation of Universal Dependencies (UD) guidelines for the annotation of numerals in Portuguese. We show the importance of knowing the topic to be instantiated in the context of several languages, since UD annotation aims to promote, as much as possible, parallelism between languages. We then explore the description of numerals in grammars, seeking subsidies to elaborate instructions for assignment of the UD part of speech tag NUM. The results of combining the UD guidelines with the characteristics of numerals in Portuguese are presented in detail, with examples, along with arguments supporting each decision made.*

Resumo. *Este artigo descreve a instanciação das diretrizes do modelo Universal Dependencies (UD) para a anotação de Numerais em português. Mostramos a importância de conhecer o tópico a ser instanciado no contexto de várias línguas, uma vez que a anotação UD tem por objetivo promover, o máximo possível, o paralelismo entre as línguas. Exploramos então a descrição dos numerais nas gramáticas, buscando subsídios para elaborar instruções para atribuir a etiqueta NUM da UD. Os resultados da combinação das diretrizes da UD com as características dos numerais em português são apresentados em detalhes, com exemplos, juntamente com os argumentos que amparam cada decisão tomada.*

1. Introdução

Universal Dependencies (UD) é uma iniciativa multinacional de anotação sintática de córpus que tem por objetivo estabelecer um maior paralelismo entre as línguas. A ideia subjacente é a de que, usando um mesmo esquema de anotação, facilita-se o estudo e a construção de aplicações multilíngues de processamento de línguas naturais. Por "esquema de anotação", entende-se tanto conjuntos comuns de etiquetas (morfológicas, morfossintáticas e de relações de dependência sintática) quanto diretrizes comuns para atribuição dessas etiquetas. O projeto UD é uma abordagem lexicalista (as palavras são as unidades mínimas às quais se atribuem as etiquetas) e fundamentada em gramáticas de dependências (Nivre et al., 2020). Atualmente o site da UD[1] disponibiliza cerca de 200 córpus anotados em mais de 100 línguas, o que demonstra a grande adoção do

[1] https://universaldependencies.org/

modelo. Das línguas do Brasil, no momento da escrita deste artigo, há três córpus de português disponíveis (córpus PUD, GSD e Bosque-UD) e seis córpus de línguas indígenas (apurina, guajajara, kaapor, makurap, mundukuru e tupinambá).

Um desafio que se apresenta para todas as línguas antes de fazer a anotação é adaptar as diretrizes da UD para as especificidades da língua. Essa fase de um projeto de anotação de córpus é chamada por Hovy e Lavid (2010) de "instanciação da teoria". A UD recomenda que, uma vez que o trabalho de instanciação da teoria tenha sido concluído, as diretrizes específicas da língua sejam publicadas no próprio site da UD, de forma que as demais iniciativas de anotação possam tomá-las como base. Até onde é de nosso conhecimento, ainda não foram publicadas diretrizes específicas para atribuição das etiquetas da UD para o português. É muito desejável que essa lacuna seja sanada para que alcancemos anotações consistentes em UD, não apenas dentro de um mesmo córpus, mas entre os córpus de português. Material teórico de suporte à anotação específico da língua é um recurso essencial para que eventuais anotadores de córpus tomem decisões menos subjetivas (na medida do possível) quando se deparam com situações não previstas nas diretrizes gerais da UD.

Imbuídos do propósito de prover os anotadores de UD em português com diretrizes de anotação, decidimos divulgá-las sob a forma de manuais de anotação. Nesses, cada uma das 17 *PoS tags* (*Part-of-Speech tags* ou etiquetas morfossintáticas) e das 37 *deprel* (*dependence relations* ou etiquetas de relações de dependência) da UD serão objeto de instruções detalhadas, ricamente ilustradas com exemplos. Neste artigo, descrevemos parte do esforço de elaborar esse material. Abordamos a questão da anotação morfossintática dos numerais em português de acordo com as diretrizes da UD e os estudos e argumentos que embasaram nossas decisões.

O artigo está organizado em 5 seções, incluindo esta introdução. Na Seção 2, apresentamos reflexões acerca dos numerais e sua expressão linguística. Em seguida, na Seção 3, expomos as divergências que encontramos entre gramáticos no que concerne à classificação dos numerais no português. Na Seção 4, descrevemos as diretrizes que produzimos para a anotação de numerais em português seguindo a abordagem da UD e, por fim, na Seção 5, tecemos nossas considerações finais.

2. Os numerais como parte integrante das línguas naturais

Os sistemas numéricos (ou sistemas de contagem) têm uma quantidade finita de algarismos únicos. Por exemplo, os algarismos arábicos constituem um conjunto com 10 integrantes (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) e os algarismos romanos constituem um conjunto com 7 integrantes (as letras I, V, X, L, C, D e M). A partir desses algarismos únicos, os sistemas de contagem utilizam estratégias de posicionamento, de soma, de multiplicação e de subtração para expressar outras quantidades. Assim, um "1" na segunda casa decimal representa uma dezena e, somado a um "3" na primeira casa, resulta no numeral 13. Nos algarismos romanos, além de multiplicação e adição, utiliza-se também a operação de subtração, motivo pelo qual "quarenta" é expresso por XL (L menos X, ou seja, cinquenta menos dez).

Na escrita, um token de numeral pode conter, além de vários algarismos, sinais de pontuação, como vírgula, ponto e barra: 1,07; 127.234; 4/7. No entanto, quando um numeral é expresso por extenso, nem sempre corresponde a um único token: 8 é "oito" (um *token*), mas 349 é "trezentos e quarenta e nove" (cinco *tokens*). Há também casos em que um número constituído de mais de um algarismo corresponde uma única palavra, como 15 (dois algarismos), que é expresso como "quinze" (um *token*). Em suma, há falta de simetria entre a quantidade de algarismos de um número e a quantidade de palavras usadas para expressá-lo por extenso.

Além disso, outra assimetria é observada: línguas que compartilham um mesmo sistema numérico possuem quantidades diferentes de itens lexicais para expressar os números. Apesar dessas diferenças, Hurford (2007) defende que há regras universais atuando na forma como as línguas usam um conjunto finito de palavras para expressar uma série infinita de números. Para ele, as estratégias usadas para compor os números a partir de uma quantidade finita de algarismos são as mesmas usadas na expressão linguística desses mesmos números. As principais estratégias são a junção (soma) e a multiplicação. No Português, a junção de algarismos é comumente indicada pela coordenação de palavras, usando uma conjunção aditiva, enquanto a multiplicação é indicada pela justaposição de palavras. Assim, 37 (30 + 7) é "trinta *e* sete" e 8.000 (8 x 1000) é "oito mil". Outras línguas, contudo, utilizam a justaposição em ambas as situações. Há, inclusive, línguas que concatenam as palavras em outra ordem, como em alemão, na qual 37 é siebenunddreißig (sieben=7 und=e dreißig=30).

O fato de cada língua possuir uma quantidade diferente de palavras para expressar os números prejudica um pouco o paralelismo entre elas. No francês, por exemplo, setenta, oitenta e noventa não são números expressos por palavras únicas: 70 é soixante-dix (sessenta mais dez), 80 é quatre-vingt (quatro vezes vinte) e 90 é quatre-vingt-dix (quatro vezes vinte, mais 10). Da mesma forma, no inglês e no francês, ao contrário do português, não há palavras para designar as centenas e, por isso, são usadas combinações da palavra "cem" com a quantidade de centenas, ficando subentendida a multiplicação. Assim, 200 é *two hundred* (dois cem) em inglês e *deux cent* (dois cem) em francês . Essas poucas comparações são suficientes para se perceber que é um desafio, para a UD, manter o máximo de paralelismo entre as línguas no que diz respeito à anotação de numerais.

3. Os numerais nas gramáticas do português

Em 1959, o Ministério de Educação e Cultura publicou, com força de lei, a Nomenclatura Gramatical Brasileira (NGB). O objetivo era criar uma ontologia dos tópicos de gramática que deveriam ser ensinados no Brasil, padronizando os termos usados para designar cada tópico. A NGB estabeleceu que há 10 classes de palavras no português, uma das quais é a dos numerais. Segundo o documento, a classe dos numerais se subdivide em quatro subclasses: cardinais (um, dois, três), ordinais (primeiro, segundo, terceiro), multiplicativos (dobro, triplo) e fracionários (meio, terço).

A NGB[2] está em vigor até hoje e orientou a confecção de diversas gramáticas escolares desde 1959. Todos os gramáticos que almejavam ter suas obras referenciadas para uso no ensino e em concursos tiveram que seguir a norma estabelecida pela NGB. Curiosamente, quando Portugal publicou a Nomenclatura Gramatical Portuguesa, em 1967, apresentou as mesmas 10 classes de palavras e, na classe dos numerais, incluiu a subclasse dos "numerais coletivos", ou seja, palavras que trazem embutida a ideia de uma quantidade (quinzena, bimestre, biênio, dúzia, dezena, par, quinteto, etc.).

Desde o início da vigência da NGB, contudo, não faltaram críticas a vários de seus tópicos. Uma das críticas é justamente com relação à classe dos numerais. Câmara Jr. (1986), por exemplo, diz que a NGB misturou critérios de forma, função e sentido das palavras ao propor 10 classes gramaticais (para ele só existem três classes de vocábulos: Nomes, Pronomes e Verbos). O autor considera os numerais como pertencentes à classe dos Nomes, a qual abriga as funções de substantivo e adjetivo. Monteiro (1991) também considera os numerais como integrantes da classe dos Nomes e exemplifica: em "três é ímpar", a palavra "três" exerce a função de substantivo e, em "três ímpares", a palavra "três" exerce a função de adjetivo.

Para Azeredo (2000), numeral é uma função semântica e, por isso, não deveria haver uma classe chamada "numerais". Segundo o autor, "O numeral é sempre constituinte de um sintagma nominal, ora ocupando a posição de núcleo - numerais fracionários e multiplicativos, ora ocupando a posição de termo adjacente - numerais cardinais e ordinais." (Azeredo, 2000, versão digital).

Uma visão intermediária entre a NGB e os autores que descartam a pertinência de uma classe de numerais é apresentada por Macambira (1987). Segundo esse autor, apenas os numerais cardinais pertencem à classe dos numerais propriamente ditos. Os numerais ordinais, segundo ele, comportam-se mais como os adjetivos, ao passo que os fracionários e multiplicativos comportam-se como substantivos. Além disso, apenas numerais que modificam diretamente um substantivo devem ser considerados, o que implica a exclusão de palavras quantitativas que se ligam ao substantivo por intermédio da preposição "de". Assim, em "mil canetas", "mil" é numeral, enquanto em "um milhão de canetas" "um" é numeral e "milhão" é substantivo. Macambira (1987) destaca também que apenas o número "um" e o feminino "uma" são singulares, pois os demais números, por expressarem mais de um, trazem implicitamente a ideia de plural. Com isso, o autor conclui que nenhum numeral admite a flexão de número, pois não há sentido em fazer o plural de uma palavra que já é plural. Esses critérios de Macambira para julgar se uma palavra pertence ou não à classe dos numerais serão, como veremos na Seção 4.4, muito relevantes dentro da abordagem de anotação UD.

É importante esclarecer que, embora nenhum numeral admita flexão de número, na língua portuguesa há onze numerais que admitem flexão de gênero:

[2] Os bastidores da criação da NGB são discutidos por Henriques (2009), o qual traz anexas a Nomenclatura Gramatical Brasileira e a Nomenclatura Gramatical Portuguesa.

> meio/meia, um/uma, dois/duas, duzentos(as), trezentos(as), quatrocentos(as), quinhentos(as), seiscentos(as), setecentos(as), oitocentos(as) e novecentos(as).

4. A anotação de numerais segundo as diretrizes da UD

As diretrizes da UD apresentam muitas diferenças em relação ao que está previsto para os numerais na NGB e na NGP. Há uma *PoS tag* específica para numeral na UD (NUM), porém ela é reservada unicamente aos numerais cardinais, seja em sua forma como algarismos (arábicos ou romanos), seja por extenso. Os numerais ordinais, por outro lado, são anotados na UD como adjetivo (ADJ). Assim, **9** é NUM e **9°** é ADJ. As demais subclasses de numerais previstas na NGB e na NGP (multiplicativos, fracionários e coletivos) não são mencionadas pela UD.

Para a UD, a classe dos numerais (representada pela *PoS tag* NUM) é uma classe que abriga palavras de uma classe fechada (os algarismos e suas respectivas formas por extenso), que ora exercem função de modificador nominal, ora função de pronome, e ora função de substantivo:

> Ele teve **três** chances de acertar. As **três** foram desperdiçadas. E achava que o **três** era seu número de sorte!

Na primeira sentença, "três" atua como modificador nominal; na segunda, como pronome e na terceira, como substantivo. Em todas as sentenças, contudo, "três" é anotado como NUM. Assim, os tokens em negrito a seguir são anotados com NUM:

> Nós **dois** somos brasileiros.
> Há **2** anos que nos conhecemos.
> São **dois** os motivos pelos quais desistimos.
> **Dois** de nós são brasileiros e **dois** são estrangeiros.
> **Dois** mais **dois** são quatro.
> O Canal **2** é a TV Cultura.

Apresentamos, em seis subseções, as principais questões que suscitam dúvidas durante a anotação de numerais.

4.1 Frações

As frações não são sinônimo de numerais fracionários. Os numerais fracionários da NGB correspondem aos denominadores das frações, quando expressos por uma única palavra. Na UD, o numerador de uma fração é sempre expresso por um numeral cardinal e o denominador é expresso por um substantivo (de "meio" a "décimo") ou por um cardinal seguido do substantivo "avos": 3/15 = três quinze avos (a partir do denominador 11: onze avos). Embora os denominadores de "quarto" a "décimo" tenham formas iguais aos numerais ordinais (anotados como ADJ), eles devem ser anotados como NOUN nessa função.

> ⅔ = dois terços (NUM, NOUN)
> ¾ = três quartos (NUM, NOUN)
> 4/11 = quatro onze avos (NUM, NUM, NOUN)

A palavra "meio" (½) é a única que denota uma fração completa e, por isso, é anotada como NUM quando se liga diretamente a um substantivo, quantificando-o.

½ = meio, meia (NUM): Comeu meia pizza.

No entanto, em contexto matemático usa-se dizer "três meios" (correspondendo à fração imprópria 3/2) e, nessa situação, "meio" é NOUN, como os demais denominadores de frações, pois flexiona no plural. Cabe aqui mencionar que há outro NUM representado pela palavra "meia", que é quando o algarismo 6 é expresso por extenso como "meia", alternativamente a "seis".

4.2 Algarismos romanos

A UD, em suas diretrizes, trata os algarismos romanos da mesma forma que os algarismos arábicos, ou seja, considera ambos numerais cardinais e, consequentemente, atribui-lhes a etiqueta NUM. Contudo, é importante ressaltar que os algarismos romanos podem apresentar uma leitura de numeral ordinal em algumas situações no português e, mesmo assim, devem ser anotados como NUM e não como ADJ como os numerais ordinais. Isso porque, enquanto os numerais ordinais têm uma marca gráfica indicando sua condição, os algarismos romanos não a têm em português. É possível, inclusive, em alguns casos, admitir as duas leituras, como é o caso de "capítulo XII" (capítulo doze ou capítulo décimo segundo). Assim, nos dois exemplos abaixo, o numeral VIII é anotado como NUM, embora no segundo tenha leitura de um numeral ordinal.

No século **VIII** não se sabia que a Terra era redonda ainda. (VIII = oito)
O rei Henrique **VIII** casou-se sete vezes. (VIII = oitavo)

4.3 Palavras que expressam ideias numéricas

O fato de a UD não mencionar outras formas de numerais, além dos cardinais (NUM) e ordinais (ADJ), não é estranho. Como vimos, os críticos da NGB diziam que os numerais pertencem à classe dos nomes, a qual abriga as funções de substantivo e adjetivo. Por exemplo, entre os multiplicativos estão incluídos os substantivos "dobro" e "triplo" e os adjetivos "duplo" e "tríplice"; entre os coletivos, estão incluídos os substantivos "bimestre" e "biênio" e os adjetivos "bimestral" e "bianual". Além disso, há os prefixos que permitem formar novos substantivos: "bi", "di", "tri", etc. Algumas palavras podem, inclusive, ora ser ADJ, ora ser NOUN, dependendo do contexto:

O DNA é uma cadeia de **dupla** hélice. (dupla = ADJ)
A **dupla** sertaneja foi formada nos anos 90. (dupla = NOUN)

4.4. O caso de "cento", "milhão", "bilhão"

Há algumas palavras que, por participarem da expressão de um número, podem causar estranheza por serem anotadas como substantivos (NOUN). É o caso de "cento", "milhão", "bilhão", "trilhão", etc.

101 = cento e um
2.221.000 = dois milhões e duzentos e vinte e um mil

Contudo, tais palavras nunca se ligam a um substantivo sem a intervenção da preposição "de", ao contrário dos numerais, que se ligam diretamente como modificadores nominais. É agramatical dizer:

***Cento** pessoas compareceram.

*__Milhões__ pessoas compareceram.

"Cento" é uma palavra que substitui "cem" nas expressões de números de 101 a 199, ou seja, num contexto bem limitado. Há contextos, porém, em que o comportamento de "cento" mostra claramente tratar-se de um substantivo, sendo passível de flexão de número e ligado a outro substantivo por meio da preposição "de":

Dois **centos** de docinhos

No caso de "milhão" (assim como "bilhão", "trilhão", etc.), percebe-se que é palavra flexionável em número e exige a preposição "de" para se ligar a um substantivo. Quando precedida de um verdadeiro numeral, "milhão" denota quantidade precisa:

Dois **milhões** de pessoas compareceram.

Quando utilizada no plural e não precedida de um verdadeiro numeral, "milhão" denota uma quantidade imprecisa, genérica:

Milhões de pessoas compareceram.

Portanto, os critérios de Macambira (1993), apresentados na Seção 3, mostram-se relevantes para distinguir um numeral (NUM) de um substantivo (NOUN) na UD e ratificam a interpretação de que "cento", "milhão", "bilhão", etc. devem ser anotados como NOUN.

4.5 A ambiguidade das formas "um/uma"

O caso que apresenta maior dificuldade na anotação de numerais está associado à ambiguidade das palavras "um/uma". Nem sempre é simples decidir entre as etiquetas DET (determinante, que é a categoria da UD que abriga os artigos indefinidos), PRON (pronome) e NUM. Para auxiliar os anotadores nessa decisão, fixamos algumas regras, parte das quais coincide com pistas de desambiguação mencionadas por Macambira (1987). Trata-se de NUM se:

- "um" ou "uma" responder à pergunta "Quantos?". Ex: Ganhou uma medalha de ouro. Quantas medalhas ganhou? Resposta: uma.
- "um" ou "uma" estiver em oposição a outras quantidades na mesma sentença. Ex: Comprou um abacate e três maçãs.
- "um" ou "uma" estiver seguido da preposição "de", indicando a seleção de um entre vários, mas que poderia ser "dois", "três", etc.. Ex: **Um** dos alunos foi eleito representante.[3]

Trata-se de DET se:

- o substantivo que sucede "um" e "uma" for incontável. Ex: Tenho **uma** grande esperança de que isso funcione.
- "um" ou "uma" puder ser suprimido. Ex: Tive **uma** impressão errada. Tive impressão errada.

[3] Observa-se que a sentença não muda de sentido com a inversão dos constituintes: "Dos alunos, **um** foi eleito representante". Além disso, "um" poderia ser substituído por "dois" ou outro numeral: "dois dos alunos foram eleitos representantes".

- "um" ou "uma" não puder ser substituído por outros números. Ex: Saiu para dar **um** passeio. *Saiu para dar **dois** passeios (não é agramatical, mas é improvável)

Trata-se de PRON se:

- "um" ou "uma" estiver em oposição a outros pronomes indefinidos, como "outro" e "outra". Ex: **Um** ou **outro** vai vencer.
- "um" ou "uma" estiver precedido do pronome indefinido "cada" (DET na UD). Ex: **Cada um** deu o melhor de si. **Cada um** de nós deu o melhor de si.[4]

4.6 Casos especiais de tokens que incluem numerais

Na UD, problemas de tokenização podem alterar a forma de anotação. Se um numeral cardinal fizer parte de um token que contém um substantivo ou um nome próprio, por exemplo, ele não será anotado como NUM, mas sim com a etiqueta da outra classe:

> A banda **U2** fez muito sucesso no Brasil.
> (U2 é anotado com a etiqueta de nome próprio PROPN).
>
> Hoje é **24/jul/2021**
> (como contém o nome do mês, esse token é anotado na UD como substantivo, ou seja, um NOUN).
>
> Já andamos **24km** nesta semana.
> (24km é anotado como substantivo, pois contém uma unidade de medida e unidades de medida, abreviadas ou por extenso, são NOUN, segundo a UD).

6. Considerações finais

Apresentamos o desafio de instanciar as diretrizes da UD na língua portuguesa no que se refere à anotação morfossintática dos numerais. Esperamos, com isso, auxiliar outros estudos linguísticos e projetos de anotação de córpus que se filiem ao mesmo modelo, pois poderão antecipar questões relevantes. No nosso caso, embora tenhamos elaborado um manual prévio para orientação dos anotadores, tivemos que complementá-lo para contemplar situações que geraram dúvidas durante a anotação. Também produzimos instruções para as demais etiquetas utilizadas pela UD e, em breve, esperamos divulgar o Manual para Anotação de *PoS tags* da UD na Língua Portuguesa, assim como produzir outros artigos com descrições e discussões linguísticas relevantes.

O trabalho reportado faz parte de um projeto maior - o POeTiSA[5], cujo propósito é avançar as pesquisas em sintaxe e *parsing* para o português brasileiro, construindo um grande treebank multi-gênero e produzindo *taggers* e *parsers* do estado da arte.

Agradecimentos

Este trabalho foi realizado no âmbito do Centro de Inteligência Artificial da USP (C4AI - http://c4ai.inova.usp.br/), com o apoio da IBM e da FAPESP (#2019/07665-4).

[4] Aqui a regra do "de" é sobreposta pela regra do "cada". Na expressão "cada um de", não há opção de inverter a ordem dos constituintes: *Dos alunos, cada **um** deu o melhor de si.
[5] https://sites.google.com/icmc.usp.br/poetisa

Referências bibliográficas

Azeredo, José Carlos de. Fundamentos de Gramática do Português. Editora Zahar, 2000. Versão eletrônica.

Câmara Jr., Joaquim Mattoso. Dicionário de Linguística e Gramática: referente à língua portuguesa. Petrópolis: Vozes, 1986, 13ª ed.

Henriques, Claudio Cezar. Nomenclatura Gramatical Brasileira: cinquenta anos depois. São Paulo: Parábola, 2009.

Hovy, E. and Lavid, J. (2010). Towards a 'Science' of Corpus Annotation: A New Methodological Challenge for Corpus Linguistics. International Journal of Translation, Vol. 22, N. 1, pp. 13-36.

Hurford, James R. (2007). A performed practice explains a linguistic universal: Counting gives the Packing Strategy. Lingua,Volume 117, Issue 5, p. 773-783.

Macambira, José Rebouças. A Estrutura Morfo-Sintática do Português. São Paulo: Livraria Pioneira Editora, 1987.

Monteiro, José Lemos. Morfologia Portuguesa. Campinas: Pontes, 1991.

Nivre, J.; Marneffe, M-C.; Ginter, F.; Hajič, J.; Manning, C.D.; Pyysalo, S.; Schuster, S.; Tyers, F.; Zeman, D. (2020). Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In the Proceedings of the 12nd International Conference on Language Resources and Evaluation (LREC), pp. 4034-4043.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Construções de Estrutura Argumental com Argumento Preposicionado: uma modelagem linguístico-computacional na FrameNet Brasil

Vânia Gomes de Almeida[1], Tiago Timponi Torrent[2]

[1]Doutoranda em Linguística - Universidade Federal de Juiz de Fora

[2]Professor Associado – Universidade Federal de Juiz de Fora – Juiz de Fora, MG - Brasil

vania.almeida@estudante.ufjf.br, tiago.torrent@ufjf.edu.br

Resumo. *Este trabalho apresenta uma proposta para representar computacionalmente as construções Transitiva Indireta, Transitiva Oblíqua e Bitransitiva do Português Brasileiro, no âmbito do Constructicon da FrameNet Brasil. Dessa forma, demonstra de que maneira as teorias irmãs da Semântica de Frames e da Gramática das Construções podem contribuir na busca por um modelo de língua que alie forma e sentido em uma única representação.*

Abstract. *This work presents a proposal to computationally represent the Indirect Transitive, Oblique Transitive and Ditransitive constructions of Brazilian Portuguese, in the framework of the Constructicon of FrameNet Brasil. In this way, it demonstrates how the sister theories of Frame Semantics and Construction Grammar can contribute to the quest for a language model bringing together form and meaning in one same representation.*

1. Introdução

Este trabalho tem como objetivo a apresentação de uma proposta de investigação de uma modelagem linguístico-computacional de construções de estrutura argumental no Português Brasileiro (PB). Tal proposta se enquadra nas discussões teórico-metodológicas da FrameNet Brasil (FN-Br) (SALOMÃO, 2009). A FN-Br está envolvida em vários campos de pesquisa, que incluem desde a construção de recursos lexicais à identificação e catalogação de construções. A FrameNet começou em 1997 como um projeto de lexicografia computacional para o inglês, liderado por Charles J. Fillmore no *International Computer Science Institute*, em Berkeley, Califórnia. Desde então, o recurso original vem expandindo sua cobertura de unidades lexicais para outros idiomas, como espanhol, alemão, sueco, japonês e português do Brasil. Nos projetos afiliados à FrameNet de Berkeley, os frames e Elementos de Frames são expandidos para os outros idiomas, ou seja, é utilizada a estrutura definida para o inglês como ponto de partida para a descrição do idioma de destino com adaptações e mudanças na estrutura original.

Diante disso, a FN-Br conta com dois grandes recursos computacionais: um *Lexicon* e um *Constructicon*. O Lexicon tem por objetivos: criar uma representação computacional de *frames*, definidos por seus participantes e instrumentos e conectados entre si via relações entre *frames*; definir Unidades Lexicais (ULs), pareamentos entre um lema e um significado definido em termos de um *frame*; anotar sentenças que exemplifiquem os padrões de valência sintáticos e semânticos em que as ULs ocorrem. Como frames são representações de estruturas cognitivas, definidas, de acordo com

Fillmore (1982), como um sistema de conceitos relacionados de tal forma que, para entender qualquer um deles, é necessário tem que entender toda a estrutura na qual ele se encaixa, as análises realizadas para as ULs, portanto, nos fornecem a descrição de suas propriedades de valência sintática e semântica

Já o *Constructicon* visa à criação de um recurso para a descrição das características semânticas e gramaticais de construções do PB, incorporando descrições interpretáveis computacionalmente para cada construção, oferecendo informações semânticas e especificando as relações entre as construções. *Lexicon* e *Constructicon* se encontram conectados porque ambos descrevem satisfatoriamente as principais propriedades dos fenômenos linguísticos, na medida em que itens lexicais são construções e são, portanto, licenciados por construções lexicais, e que tanto itens lexicais quanto construções não lexicais podem evocar *frames,* que constituem o cerne de uma FrameNet.

Diante da interligação entre as duas frentes da FN-Br, apresentamos uma proposta que pretende ampliar o que foi apresentado em Almeida (2016) e em Diniz da Costa *et al.* (2018), em que buscou-se apresentar a modelagem de um conjunto de 11 construções de estrutura argumental do PB de modo a aprimorar o *Constructicon* da FN-Br, sendo elas: Sujeito_predicado, Transitiva_direta_ativa, Intransitiva, Ergativa, Predicativa_nominal, Predicativa_nominal_atributiva, Predicativa_nominal_estativa, Predicativa_locativa, Mudança_de_estado, Argumento_cindido e Objeto_interdito. Mais especificamente, debruçamo-nos sobre três construções de estrutura argumental que possuem argumentos preposicionados: Transitiva_indireta, Transitiva_oblíqua e Bitransitiva.

2. A Gramática de Construções

A Gramática de Construções (GC) é um desenvolvimento recente na teoria linguística, começou a ser discutida mais intensamente no final dos anos 80 e início dos anos 90 com os trabalhos de Fillmore, Kay e O'Connor (1988), Goldberg (1995), Kay e Fillmore (1999) e Croft (2001).

Nesse contexto, as principais acepções da GC são: construções são unidades básicas da língua que se constituem em correspondências entre forma e significado (GOLDBERG, 1995; KAY e FILLMORE, 1999); qualquer material linguístico desde o mais simples ao mais complexo pode ser considerado uma construção, havendo uma continuidade entre o léxico e a sintaxe; a gramática de uma língua consiste de uma rede de construções mediada por diferentes relações (FILLMORE, 2008).

Há diversas abordagens que utilizam essas acepções básicas da GC, sendo talvez o modelo mais conhecido o apresentado por Goldberg (1995, 2006), que propõe um modelo baseado na interação entre semântica verbal e a semântica construcional. Com esse modelo, não é necessário apresentar vários sentidos verbais para explicar a diferença entre as construções de estrutura argumental.

O modelo de Goldberg (1995) para a GC desenvolveu-se em uma produtiva interface com a área de aquisição da linguagem (GOLDBERG, 2006). Esse modelo, entretanto, possui limitações de formalização, que tornam difícil sua aplicação para além da fase linguística do processo de modelagem computacional[1]. Nesse sentido, o modelo

[1] Dias da Silva (2006) apresenta três fases como necessárias para o desenvolvimento de um sistema em Linguística Computacional: a fase linguística, a fase linguístico-computacional e a fase computacional. Neste trabalho, a primeira apresenta a especificação das construções, a segunda, a representação dessas construções no Constructicon e a fase computacional propõe a projeção de um sistema computacional para que essas construções possam ser reconhecidas posteriormente.

da GC baseado em unificação proposto por Fillmore, Kay e O'Connor (1988), por Kay e Fillmore (1999), Fillmore (2008) e Fillmore (2013), a Gramática de Construções de Berkeley (BCG), revela-se mais adequado para o objeto deste trabalho.

O que a BCG traz de diferente de outros modelos de GC é o fato de ser um modelo de gramática que integra os padrões sintáticos e semânticos das expressões linguísticas às noções da Semântica de Frames (FILLMORE, 1982; 1985), fornecendo um formalismo baseado em unificação, que é relevante para a proposta de desenvolvimento de recursos como *Constructicons* e para tarefas de Compreensão de Língua Natural.

3. O *Constructicon* da FrameNet Brasil

Seguindo o proposto na BCG (FILLMORE, 2013), o *Constructicon* foi criado com o intuito de representar computacionalmente determinadas estruturas linguísticas não processáveis lexicograficamente e, por isso, abarca o conhecimento linguístico que excede a valência simples de palavras simples (FILLMORE, 2008). De modo mais específico, descreve construções em termos de suas propriedades gramaticais e seu potencial semântico. A iniciativa de criação de um modelo computacional de construções foi seguida também pela FN-Br que, desde 2010, desenvolve um *Constructicon* para o PB (TORRENT ET AL. 2014; SILVA ET AL., 2017).

No *Constructicon*, uma construção é mapeada formalmente, unificando-a, quando relevante, a um *frame* específico. Portanto, para desenvolver o *Constructicon* de acordo com os princípios da Semântica de Frames e da Gramática de Construções, foi necessário: (i) fornecer um meio de ligar construções a *frames* da mesma maneira que Unidades Lexicais estão conectadas a *frames*, (ii) modelar relações entre construções, de modo que o recurso possa fornecer como resultado uma rede de construções, e não uma lista delas, (iii) definir restrições sintático-semânticas que possibilitem a descrição dessas construções computacionalmente.

Diante disso, o *Constructicon* conta hoje com dois tipos de relações – Herança e Evocação – e com restrições – *Constraints* – de modo a apresentar linguística e computacionalmente as principais características de construções no PB.

A primeira relação, Herança construcional, conecta uma construção à sua construção mãe, ou seja, ela é modelada seguindo Kay e Fillmore (1999). Nesse contexto, o modo completo de herança presume que a construção herdeira é um tipo mais específico da construção mãe. Essa relação fornece a possibilidade de armazenar informações de modo econômico e não redundante, tratando a herança pelo viés da cópia virtual de informação, isto é, a informação herdada é registrada somente na construção dominante, o que evita a redundância no armazenamento de dados. Assim, a construção filha contém todas as informações da construção mãe e mais outras informações adicionais.

A segunda relação, Evocação, conecta uma construção a um ou mais *frames* que ela pode evocar. Diferentemente da Herança, a relação de Evocação não implica que o *frame* evocado seja "mãe" da construção que o evoca, isto é, essa não necessita conter toda a estrutura de seu esquema "mãe". Assim, o *frame* evocado pela construção é evidenciado através do mapeamento entre os Elementos de Construção e os Elementos de Frame que a construção evoca.

As construções no *Constructicon* da FN-Br também são definidas em termos de restrições ou *constraints*. Na base de dados, uma restrição é uma relação entre um Elemento de Construção e qualquer Entidade, como um *Frame*, uma Construção ou um

Elemento de Construção. Na próxima seção, serão demonstrados exemplos e aplicações das restrições bem como das relações de Herança e Evocação.

4. Construções de Estrutura Argumental no PB

As construções denominadas de estrutura argumental ganharam destaque principalmente nos trabalhos de Goldberg (1995, 2006). Segundo a autora, as construções de estrutura argumental são compostas por predicadores e seus respectivos argumentos. A proposta da GC nos permite associar uma estrutura sintática a um significado específico, uma vez que as construções de estrutura argumental representam ações humanas básicas e licenciam sentenças independentes de verbos particulares.

As construções analisadas nesse trabalho são de tipologia Sujeito – Verbo – Objeto (SVO) de período simples. Optamos por tratar dessas construções já que essas já foram descritas por diferentes linguistas em diferentes trabalhos, e por serem estruturas representativas de construções esquemáticas.

As análises que serão apresentadas nesta seção dizem respeito ao um conjunto de construções que tomam argumentos preposicionados em PB. Primeiramente, será apresentada a descrição de cada uma delas conforme as suas características sintáticas, semânticas e, juntamente, teremos a modelagem dessas construções no C*onstructicon* da FN-Br. No total, serão apresentadas três construções – Transitiva Indireta, Transitiva Oblíqua e Bitransitiva.

4.1. Construção Transitiva Indireta

A partir das descrições realizadas por Castilho (2010), o qual apresenta estudos sobre a Construção Transitiva Indireta, propomos uma modelagem linguístico-computacional para essa construção. Castilho (2010) define que a construção Transitiva Indireta é composta por um argumento externo sujeito e um argumento interno objeto indireto, o qual seria proporcional a um pronome dativo, como *lhe*. As propriedades sintagmáticas e funcionais dessa construção também são exploradas pelo autor, assim, seria estruturada pela sequência [SN[SV[SP]]] em que o sintagma nominal designa o sujeito e o sintagma preposicionado designa o objeto indireto ou beneficiário, conforme (1) e (2).

(1) O menino obedeceu à mãe.
(2) O prefeito respondeu ao jornalista.

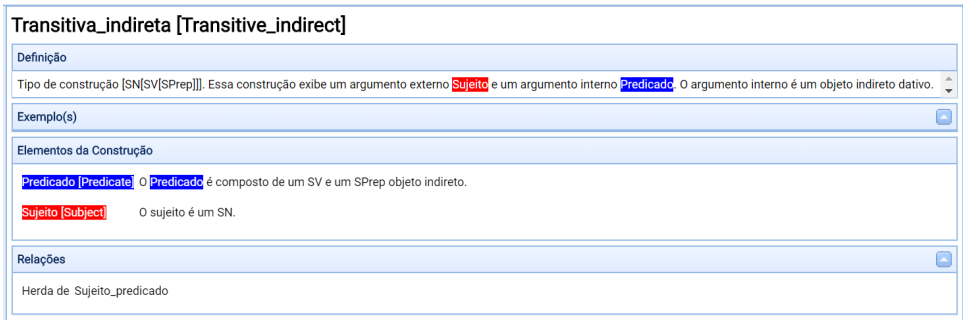


Figura 1. Construção Transitiva Indireta

O processo de modelagem da Transitiva Indireta iniciou-se a partir da definição das suas propriedades sintáticas externas, o que demonstrou ser ela composta por dois Elementos de Construção (ECs), sendo um o Sujeito e o outro o Predicado, conforme Figura 1.

Outro aspecto relevante para o tratamento de construções diz respeito à modelagem de suas restrições de preenchimento. Assim, é possível estabelecer restrições de constituência para a construção a partir da definição dos ECs, uma vez que a constituência tipifica os signos filhos – os ECs – da construção em termos de outras construções. Dessa forma, enquanto os aspectos formais são considerados para a criação dos ECs, ou signos filhos, a informação semântica é atribuída através da unificação da construção com um *frame*. A aplicação das restrições à construção em questão é mostrada na Figura 2.

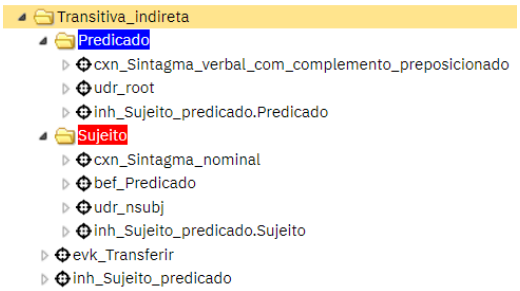


Figura 2. Restrições aplicadas aos CEs da construção Transitiva Indireta

Podemos observar na Figura 2 que o EC Predicado constitui uma construção de Sintagma verbal com complemento preposicionado, enquanto o EC Sujeito é constituído pela construção Sintagma nominal determinado.

Outras restrições que podem ser observadas na modelagem da Construção Transitiva Indireta derivam do comportamento dos ECs, que, na Figura 2 aparecem como CE. São elas: CE_*before*, determinando que um dado EC deve vir antes de outro na sentença; *CE* > *Inheritance* > *CE* indica a relação de herança entre construções e a relação de herança entre ECs; *CE* > *UDRelation* determina qual é a relação sintática do EC de acordo com as *Universal Dependencies*. No caso em tela, indicam que o Sujeito deve vir antes do Predicado; que o EC_Predicado é herdeiro do Predicado da Construção Sujeito_Predicado, o mesmo acontecendo analogamente com EC_Sujeito; e que o EC Predicado recebe a etiqueta root, por ser o núcleo dessa construção, enquanto o EC Sujeito recebe a etiqueta nsubj, indicando que ele é o sujeito dessa construção.

Além da constituência, uma operação denominada unificação, nos permite relacionar os ECs de uma construção ao *frame* por ela evocado, uma vez que os ECs da construção são relacionados aos EFs desse *frame*. Assim, o conjunto de propriedades de uma construção recebe o mesmo tratamento tanto sintático, como semântico. Conforme a modelagem realizada no *Constructicon*, a Construção Transitiva Indireta foi associada ao frame Transferir que consiste em um Doador realizando a transferência de um Tema para um Receptor.

4.2. Construção Transitiva Oblíqua

Há uma grande discussão na literatura gramatical a respeito da diferença entre o

complemento oblíquo e o objeto indireto. A razão para essa discussão perpassa tanto questões morfossintáticas como questões semânticas. Nesse trabalho, optamos por seguir principalmente as considerações de Castilho (2010), assim como apresentamos a perspectiva diversa defendida por Rocha Lima (2007), que julga essa distinção principalmente pela natureza da preposição que encabeça o Sintagma Preposicionado. Segundo este último autor, o SP será um complemento dativo ou objeto indireto quando esse for por natureza [+humano ou +animado] introduzido pela preposição a com papel semântico de fonte ou alvo. Por outro lado, o sintagma preposicionado será um complemento oblíquo quando introduzido pelas preposições *a, de, em, com* e *para* com papéis semânticos de tempo, locativo e companhia.

Castilho (2010) mostra a Construção Transitiva Oblíqua como sendo composta por um argumento externo e um argumento oblíquo, que, segundo o autor, é comumente confundido pela Gramática Tradicional como o objeto indireto. A estrutura sintagmática e funcional dessa construção é [SN [SV [SP]]], em que o primeiro SN é o sujeito e o argumento oblíquo corresponde a um sintagma preposicional como em (3) e (4):

(3) Luís gosta de peras.
(4) Pedro precisa de notas.

Nos exemplos (3) e (4) temos os construtos licenciados pela Construção Transitiva Oblíqua. Em (3) o SN *Luís* corresponde ao sujeito, enquanto o SP *de peras* representa o argumento oblíquo, já em (4) o SN *Pedro* constitui o sujeito, enquanto o SN *de notas* constitui o argumento oblíquo.

Castilho (2010) chama a atenção para o fato de alguns pesquisadores classificarem o complemento oblíquo ora como adjunto adverbial ora como complemento terminativo. De acordo com Castilho (2010), tal argumento interno não pode ser considerado nem objeto direto nem objeto indireto, por não se possível comutá-lo com os pronomes "o" e "lhes", argumentando, diferentemente de Rocha Lima (2007), que apenas os sintagmas selecionados pelo verbo serão considerados argumentos. Sendo assim, o argumento oblíquo é uma exigência verbal e não uma atribuição preposicional.

A modelagem da Construção Transitiva Oblíqua no *Constructicon* mapeou que essa construção apresenta um Sujeito que corresponde a um Sintagma nominal e um Predicado que contém um verbo e um Sintagma preposicional. A modelagem mapeou também a relação de Herança com a construção Sujeito_Predicado e a relação Evocação com o frame Evento.

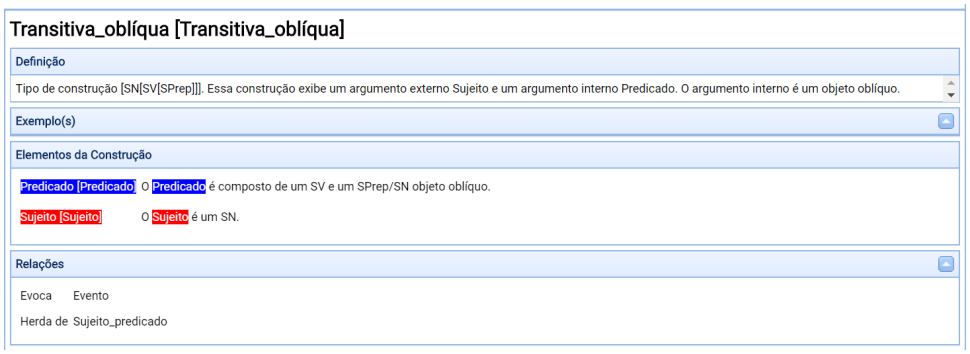


Figura 3. Construção Transitiva Oblíqua

A Figura 3 mostra a Construção Transitiva Oblíqua com os ECs correspondentes e suas respectivas relações, e a Figura 4 mostra as restrições aplicadas aos ECs da construção que são similares as da Construção Transitiva Indireta, já que em relação à constituência ambas apresentam um Sintagma verbal com complemento preposicionado, mas conseguimos mapear a diferença em relação ao *frame* que cada construção evoca, uma vez que enquanto a Transitiva Indireta evoca o *frame* Transferir a Transitiva Oblíqua evoca o *frame* de Evento.

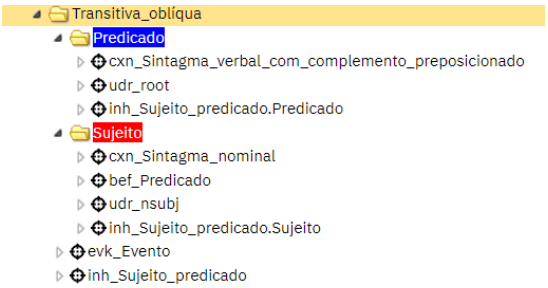


Figura 4. Restrições aplicadas aos CEs da Construção Transitiva Oblíqua

4.3. Construção Bitransitiva

A Construção Bitransitiva é também denominada triargumental. Diferente das construções biargumentais, essa construção, segundo Castilho (2010), é organizada de modo que temos um argumento externo e dois argumentos internos que podem ser representados pela estrutura sintática [SN [SV SN[SP]]]. Autores como Castilho (2010) e Perini (2010) afirmam a existência da construção em questão, uma vez que, na sua realização, temos a estrutura sintática composta por um Sujeito que corresponde a um Sintagma Nominal, outro Sintagma Nominal correspondente a um Objeto Direto e um Sintagma Preposicionado equivalente a um Objeto Indireto, como vemos em (7) e (8).

(7) João deu um livro a Pedro.
(8) O motorista colocou as malas no carro.

Na modelagem realizada no Constructicon, consideramos, portanto, as especificidades sintático-semânticas dessa construção. A composição sintática herdada da estrutura abstrata da construção Sujeito_Predicado é aquela em que o Sujeito é um SN agentivo e Predicado é um Sintagma verbal bitransitivo, composto por um SN e um SP, como podemos observar na Figura 5. Já a Figura 6 mostra a relação da Construção Bitransitiva ao *frame* por ela evocado e as restrições a ela aplicadas. Como o frame evocado pela Construção Bitransitiva é o frame Transferir que envolve um Doador transferindo um Tema para um determinado Alvo, a relação Evoca nos permite representar a contraparte semântica da construção em que o SN-Sujeito consiste no Doador e o Predicado, sendo um sintagma verbal bitransitivo composto pelas construções de Sintagma Nominal e Sintagma Preposicional, corresponde os EFs Tema e Receptor. Como vimos, a Construção Transitiva Indireta também evoca frame Transferir. Em relação ao frame, a diferença de uma construção para outra está no perfilhamento, apontado por Langacker (1991): enquanto na Transitiva Indireta o EF Tema é menos proeminente por se encontrar incorporado à raiz verbal, na Bitransitiva o Tema está

configurado no SN e apresenta o mesmo grau de proeminência que o Doador e o Receptor, tendo a Construção Bitransitiva os três EFs do frame Transferir perfilados.

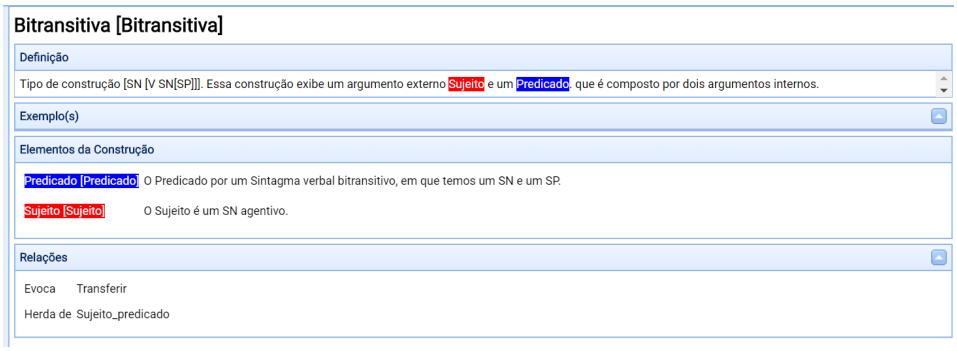


Figura 5. Construção Bitransitiva

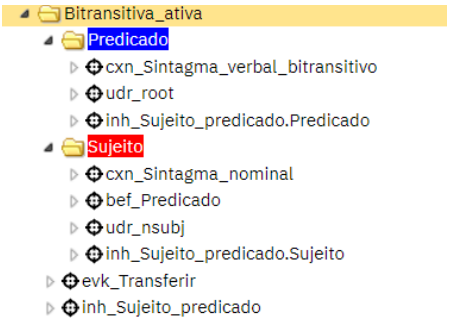


Figura 6. Restrições aplicadas aos CEs da Construção Bitransitiva

5. Considerações Finais

Neste trabalho apresentamos três construções, Transitiva Indireta, Transitiva Oblíqua. As construções que foram aqui modeladas explicitam a necessidade de considerar tanto aspectos semânticos como aspectos sintáticos para uma representação satisfatória do conhecimento linguístico. Para isso, de um lado, as restrições sintáticas foram tratadas em termos de sintagmas e categorias gramaticais como uma forma de mostrar as generalizações presentes em cada construção. Por outro lado, cada restrição foi unificada a um determinado componente semântico do *frame* evocado pela construção.

A proposta é abarcar no *Constructicon* diferentes construções de estrutura argumental com o intuito de constituir uma gramática do PB que inclua diferentes padrões construcionais, já que esses manifestam frames genéricos das ações humanas básicas, que já estão disponíveis na base da FrameNet Brasil.

O que propomos aqui ao aliar a teoria da Gramática de Construções com os recursos disponíveis da FrameNet Brasil é tentar modelar computacionalmente a cognição humana e os mecanismos que fundamentam o comportamento linguístico humano. Essa proposta tem no significado um papel crucial. Muitas vezes negligenciado em outros

formalismos, o significado, aqui, deve ser incorporado às representações gramaticais com o fim de demonstrar como um modelo baseado em construções é cognitivamente plausível.

6. Referências

Almeida, V. G. (2016). "Identificação Automática de Construções de Estrutura Argumental: um experimento a partir da modelagem linguístico-computacional das construções Transitiva Direta Ativa, Ergativa e de Argumento Cindido". Dissertação de Mestrado em Linguística. Universidade Federal de Juiz de Fora, Juiz de Fora.

Castilho, A. T. (2010). "Nova Gramática do Português Brasileiro". São Paulo: Editora Contexto.

Dias-da-silva, B. C. (2006) O estudo Linguístico-Computacional da Linguagem. In: "Letras de Hoje", v. 41, n. 2, p. 103–138.

Diniz da Costa et al. (2018) Representação computacional das construções de sujeito-predicado do português do Brasil. REVISTA LINGUÍSTICA, Rio de Janeiro, v. 14, p. 149-178.

Fillmore, C. J. (1982). Frame semantics. In: Linguistic Society of Korea (ed.), "Linguistics in the Morning Calm". Seoul: Hanshin, p.111-138.

Fillmore, C. J. (1985). Frames and the Semantics of Understanding. "Quaderni di semantica" 6 (2), p. 222- 254.

Fillmore, C. J (2008). Border conflicts: FrameNet meets construction grammar. In Bernal, E. & DeCesaris, J. (eds.). "Proceedings of the XIII EURALEX International Congress". Barcelona: Universitat Pompeu Fabra, v. 4968, p. 49-68.

Fillmore, C. J. (2013). Berkeley Construction Grammar. In: Hoffmann, T. & Trousdale, G. (eds.). "The Oxford Handbook of Construction Grammar". Oxford: Oxford University Press, p. 111-132.

Fillmore, C. J; Kay, P. & O'Connor, M. (1988). "Regularity and idiomaticity in grammatical constructions: the case of let alone". Language, 64 (3), p. 501–538.

Goldberg, A. E. (1995). "Constructions: A Construction Grammar Approach to Argument Structure". Chicago: Chicago University Press.

Goldberg, A. E. (2006) "Constructions at Work: The nature of generalization in language". Oxford: Oxford University Press.

Kay, Paul; Fillmore, Charles J. (1999). Grammatical constructions and linguistic generalizations: The 'What's X doing Y Construction? "Language", 75 (1), p.1–33.

Salomão, M. M. M. (2009). FrameNet Brasil: um trabalho em progresso. "Calidoscópio", 7(3), p. 171-182.

Silva, A. B. L. et al. (2017) In: "The AAAI 2017 Spring Symposium on Computational Construction Grammar and Natural Language Understanding Technical Report" SS-17-02. Palo Alto, CA: AAAI Publications, v.17, p.193-196.

Torrent, T. T. & Ellsworth, M. (2013). Behind the Labels: criteria for defining analytical categories in FrameNet Brasil. "Veredas", 17 (1), p. 44-65.

ROCHA LIMA, (2007) "C. H. da. Gramática Normativa da Língua Portuguesa". 46ª ed. Rio de Janeiro: José Olímpio Editora.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Modelagem de Construções Interrogativas QU- no Constructicon da FrameNet Brasil

Natália Duarte Marção[1], Tiago Timponi Torrent[1]

[1]FrameNet Brasil – Programa de Pós-Graduação em Linguística
Universidade Federal de Juiz de Fora (UFJF)
Rua José Lourenço Kelmer, s/nº, Campus Universitário
36036-900 – Juiz de Fora – Minas Gerais – Brasil

natalia.duarte@estudante.ufjf.br, tiago.torrent@ufjf.edu.br

Abstract. *This paper aims to present the description and the linguistic-computational modeling of Wh-interrogative constructions in Brazilian Portuguese in the FrameNet Brasil Constructicon.*

Resumo. *Este trabalho objetiva apresentar a descrição e a modelagem linguístico-computacional das construções interrogativas QU- do Português brasileiro no Constructicon da FrameNet Brasil.*

1. Introdução

O presente trabalho tem por objetivo apresentar a descrição e a modelagem das construções Interrogativas QU- do Português Brasileiro como resultado da pesquisa realizada por Marção (2018) durante o mestrado. Para o Português do Brasil, existem algumas análises para as construções Interrogativas QU- sob abordagens diversas. Dentre essas abordagens, destacamos as gerativistas [Adger 2003; Augusto 2005; Modesto 2012], as quais propõem que o fenômeno sintático de maior interesse de análise se relaciona ao movimento do elemento QU- em construções interrogativas em Português Brasileiro, uma vez que existe a possibilidade do movimento, mas este não é obrigatório como ocorre em outras línguas, por exemplo, o inglês.

De acordo com as abordagens gerativistas [Adger 2003; Augusto 2005; Modesto 2012], ao mover o elemento QU- para a periferia à esquerda da sentença, este deixa uma cópia em sua posição de origem, a qual não é realizada foneticamente, mas é interpretada semanticamente nesta posição. Dessa maneira, o elemento movido é realizado foneticamente, enquanto a cópia é apagada no componente fonológico. Em contrapartida, a cópia é interpretada semanticamente em sua posição de origem e não na posição ocupada após o movimento.

Focar no movimento do elemento QU- nas construções interrogativas não é suficiente para abranger todas as especificidades desse tipo de construção. Dessa forma, para nosso trabalho, as abordagens construcionistas se mostram mais adequadas, conforme se mostrará na seção que se segue.

2. Proposta Construcionista para as Construções Interrogativas QU-

Neste trabalho, recorremos à Gramática das Construções de Berkeley (*Berkeley Construction Grammar* - BCG) [Kay & Fillmore 1999; Fillmore 2013] e à Gramática das Construções Baseada em Signos (*Sign-Based Contruction Grammar* - SBCG) [Sag

2012] pelo seu caráter formal, visto que se tem por objetivo descrever e modelar construções em termos de suas propriedades gramaticais e seu potencial semântico.

Essas bases teóricas propõem que as informações sintático-semânticas de uma construção podem ser representadas através de um sistema de traços em Matrizes de Atributo e Valor (AVM). A AVM é a forma em que geralmente são apresentadas as estruturas de traços, assim, as estruturas de traços consistem na "divisão" de um dado linguístico em partes menores, os chamados atributos e a cada atributo associam-se valores.

De maneira similar, a SBCG busca formalizar descrições que sejam interpretáveis computacionalmente, integrando-as aos pressupostos fundamentais das abordagens construcionistas. Na SBCG, o pareamento de forma e significado é mediado por signos. Sag (2012) destaca que descrição de um signo incorpora traços como fonologia, forma, estrutura argumental, sintaxe, semântica e contexto e esses traços são descritos também em termos de AVMs, assim como proposto pela BCG.

Uma das principais características das estruturas de traços é poderem combinar informações através da unificação. O processo de unificação é o responsável pela aceitação ou rejeição de constituintes candidatos a ocupar posições sintáticas específicas. Portanto, a aceitação ou restrição fica condicionada à compatibilidade entre os valores dos atributos, isto é, aos elementos linguísticos e os valores exigidos pelas posições da construção. Assim, é possível definir e especificar os tipos de entidades que podem ou devem estar em cada sintagma [Fillmore 2013].

É através do processo de unificação que as AVMs combinam-se e, assim, projetam uma nova AVM que contém exatamente os valores e atributos das AVMs que se uniram [Goldberg 2006]. Logo, a principal tarefa da unificação é garantir que os atributos que possuem valores contraditórios falhem ao se combinar, de modo a não chegar a licenciar uma construção [Fried & Ostman 2004].

Sob a perspectiva da BCG, nas construções interrogativas, o elemento no início da sentença na posição mais periférica à esquerda, mantém uma ligação via unificação com uma lacuna de tal forma que, no lugar da lacuna, se mantêm as propriedades do constituinte de modo a tornar a sentença interpretável [Fillmore 1985]. Fillmore (1985) explica que os estudos gerativistas propõem que o constituinte QU- é movido para a posição inicial da sentença, além de carregar as propriedades necessárias para que ele seja interpretado na posição de origem. Ao assumir essa nova posição, esse constituinte também adquire as características exigidas por sua nova posição. Já os estudos construcionistas, segundo Fillmore (1985), se concentram nas características da posição na qual o constituinte QU- de fato ocorre.

Já a SBCG desenvolve mais a abordagem construcionista para as interrogativas QU-, que são entendidas, nesse modelo, como um tipo de construção *filler-gap*, isso porque o sintagma QU- preenche a posição inicial da sentença enquanto é deixada uma lacuna na posição em que o sintagma é interpretado. Sobre a dependência existente entre o constituinte inicial e a lacuna deixada por este, Sag (2012) afirma que pode haver uma discrepância entre os atributos e seus valores que surge da realização não-local do constituinte. De modo a formalizar essa discrepância, o autor propõe que

> a presença de uma lacuna é codificada em termos de uma especificação não vazia para o traço GAP (por exemplo, [GAP <NP>]). Por contraste, uma

> expressão que não contenha lacunas desvinculadas é especificada como [GAP <>] [SAG 2012:163, tradução nossa].

Nas construções do tipo *filler-gap*, Sag (2012), argumenta que ocorre um processo de Percolação. Nesse processo, as informações de contextos sintáticos mais baixos são passadas para contextos sintáticos superiores por restrições, logo, quando todas as AVMs filhas têm uma especificação vazia para um determinado traço, o mesmo ocorrerá com a AVM mãe. Contudo, quando uma das filhas tem uma especificação não-vazia, a mãe, então, vai carregar a mesma especificação não-vazia. Essa questão é formalizada no diagrama apresentado na Figura 1.

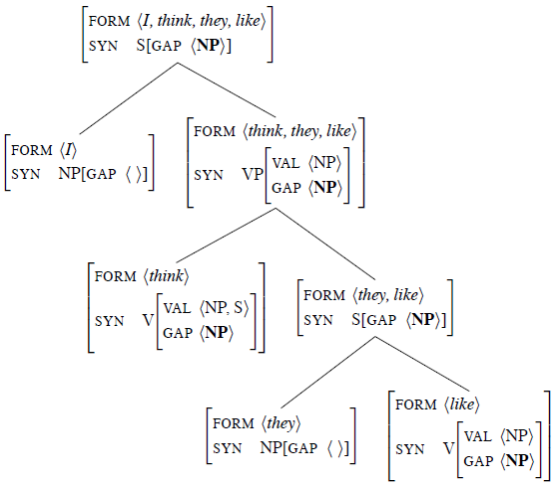


Figura 1. Representação do processo de Percolação [Sag 2012:163]

Na Figura 1, podemos ver que a especificação não-vazia para o traço GAP ([GAP <NP>]) foi "filtrada" pelas filhas e passada aos níveis superiores. Diante disso, Sag (2012) afirma que para o traço QU (WH) também ocorre o processo de Percolação. Assim, o traço QU tem um valor não-vazio que percola os níveis mais baixos das AVMs até atingir os níveis superiores.

A partir do que foi exposto, podemos dizer que a SBCG fornece uma abordagem ao tratamento das construções Interrogativas QU-, que, principalmente pela formalização proposta pelas AVMs, contribui para o tratamento computacional destas construções.

3. Constructicon da FrameNet Brasil

A partir dos pressupostos da BCG e da SBCG, o Constructicon pode ser definido como um recurso computacional sintático-semântico que contém o repertório das construções de uma língua [Fillmore 2008]. O Constructicon tem como propósito suprir a necessidade de análise de estruturas linguísticas que não são processáveis lexicograficamente, visto que, numa FrameNet, somente as valências das unidades lexicais são anotadas. O Constructicon, mais especificamente, descreve construções em termos de suas propriedades gramaticais e de seu potencial semântico. Ademais, realiza tais descrições de modo compatível com os pressupostos teóricos da BCG.

Um dos desafios de desenvolver um Constructicon, de acordo com Torrent et al. (2018), é o de representar computacionalmente as restrições e possibilidades de construções, por exemplo, como os elementos da construção se relacionam com outra construção, ou como o polo formal se relaciona com o polo semântico. Logo, é necessário que o Constructicon abarque questões relacionadas a constituência das construções, bem como, o processo de unificação.

Sobre a constituência, Fillmore et al. (2012) discorrem que, no Constructicon, cada construção é definida em termos de suas partes constituintes, os chamados Elementos da Construção (EC). Dessa forma, o processo de modelagem das construções se dá, primeiramente, pela definição das propriedades sintáticas destas construções.

Na FrameNet Brasil, o Constructicon realiza esse processo de forma mais integrada do ponto de vista da base de dados, uma vez que permite associar padrões construcionais a frames, além de permitir a adição de restrições às construções modeladas [Torrent et al. 2018]. Para ilustrar como ocorre essa integração do banco de dados, tomamos como exemplo a construção Transitiva_indireta_agentiva. Ela é constituída pelos ECs Sujeito e Predicado. Para essa construção, no processo de unificação, podemos estabelecer restrições de constituência. Dessa forma, o EC Sujeito é licenciado pela construção Sintagma_nominal_determinado, enquanto o EC Predicado é uma instância de um dos subtipos da construção Sintagma_verbal_com_complemento_preposicionado.

Ademais, o Constructicon permite estabelecer relações entre construção e frame, assim, uma construção como a Transitiva_indireta_agentiva evoca o frame de Ação_transitiva. Esse frame é basicamente definido como a ação de um Agente que afeta uma entidade, chamada de Paciente. Quando a construção evoca um frame, os ECs referem-se aos Elementos de Frame (EFs) do frame evocado pela construção. Os EFs são as partes constituintes de um frame. Sendo assim, os ECs Sujeito e Predicado da construção Transitiva_indireta_agentiva evocam, respectivamente, os EFs Agente e Paciente do frame Ação_transitiva.

As especificidades do Constructicon da FrameNet Brasil, serão mais desenvolvidas na próxima seção, na qual demonstraremos como é feita a inclusão das construções neste modelo linguístico-computacional.

4. Modelagem das Construções Interrogativas QU- no Constructicon

A modelagem de construções no Constructicon da FN-Br segue algumas etapas, como propõe Marção (2018), e estas serão apresentadas aqui.

Segundo Marção (2018), o processo de modelagem das construções Interrogativas QU- se iniciou pela modelagem de duas construções abstratas, a Interrogativa_QU e a Interrogativa_QU_preposicionada, as quais são concebidas como nós abstratos em relação às demais construções interrogativas modeladas e não podem licenciar construtos na língua. Essas construções abstratas congregam informações que são compartilhadas por todas as construções filhas, ou seja, as construções que herdam dela, além de ter a função de organizar as demais construções em rede.

Para exemplificar o processo de modelagem, Marção (2018) utiliza a construção Interrogativa_qual para demonstrar os passos que devem ser seguidos com base no emprego da ferramenta WebTool 3.0 da FrameNet Brasil.

O passo inicial da modelagem é definir os elementos que constituem a construção, i. e., os Elementos de Construção (ECs), assim sendo, a construção Interrogativa_qual é constituída pelos ECs Estrutura_argumental_base e Pronome_qual. Nesse contexto, para inserção da construção Interrogativa_qual, é necessário apontar três informações iniciais, como observado na Figura 2.

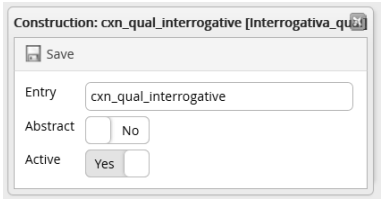


Figura 2. Edição de entrada da construção Interrogativa_qual [Marção 2018:82]

Em "Entry" é registrado o nome interlingual da construção. O nome é composto pela abreviação de construção "cnx" e o restante do nome é uma especificação da construção em questão (Interrogativa_qual). Em "abstract" sinalizamos se é uma construção abstrata ou não, no caso, está marcado "no", pois não se trata de uma construção abstrata. Por fim, em "Active" assinalamos se essa construção está ou não ativa na base de dados. Feito isso, o processo é salvo e em seguida uma nova tela se abre, a qual é apresentada na Figura 3.

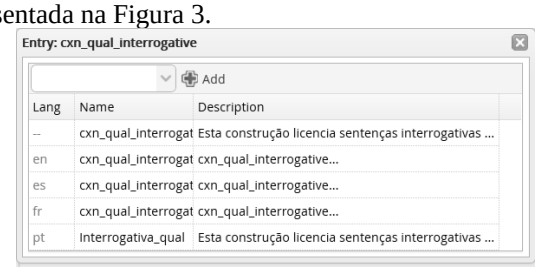


Figura 3. Edição de entrada por idioma [Marção 2018:82]

Ao clicar em alguma abreviação de idioma é possível editar a entrada de cada idioma de forma legível a humanos. Se clicarmos em "pt" somo redirecionados a outra tela (Figura 4).

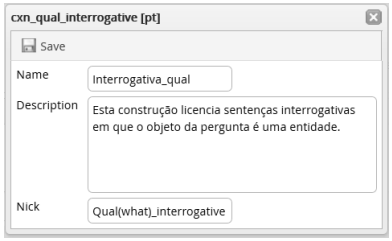


Figura 4. Edição da definição da construção em Português Brasileiro [Marção 2018:83]

Marção (2018) ressalta que na tela da Figura 4, é possível alterar o nome da construção e criar uma definição para ela em Português Brasileiro, contudo, em "Nick" é mantido o nome em inglês da construção. A partir disso, a construção foi criada e, logo, passa-se a criação e definição dos ECs.

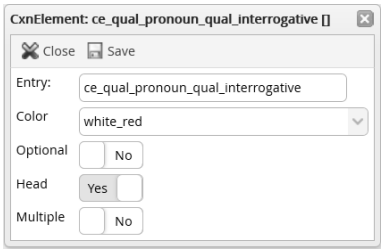


Figura 5. Edição da entrada do Elemento de Construção Pronome_qual [Marção 2018:83]

A Figura 5 apresenta a tela de edição do EC Pronome_qual. Nela insere-se o nome interlingual do EC, o qual é composto pela sigla "ce" em inglês para Elemento de Construção, "qual_pronoun" refere-se ao pronome qual, a parte "qual_interrogative" se refere à construção a qual esse EC pertence. Em seguida é possível escolher a cor em que esse elemento aparecerá, se este é um elemento opcional ou não. Também é possível assinalar se esse EC é "head" ou não, isto é, se é nuclear ou não, no caso, marcamos como nuclear. Por fim, assinalamos que o EC não é múltiplo, pois não há a possibilidade de mais de um pronome QU ser conectado recursivamente nessa construção sem que ocorra uma coordenação. Esse passo também é salvo e uma nova tela aparece (Figura 6).

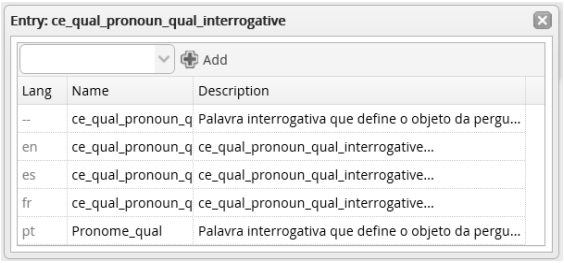


Figura 6. Edição entrada por idioma do Elemento de Construção Pronome_qual [Marção 2018:84]

A Figura 6 mostra a possibilidade de edição do EC por idioma. Clicando em "pt" abre-se uma nova janela, a qual é exibida na Figura 7. Nessa janela mantem-se o nome e cria-se a definição do EC em Português Brasileiro, contudo, em "Nick" mantemos o nome em inglês.

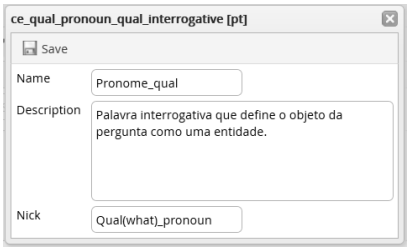


Figura 7. Edição da definição do Elemento de Construção Pronome_qual em Português Brasileiro [Marção 2018:84]

Os mesmos procedimentos são realizados para o cadastramento do EC Estrutura_argumental_base, outro EC que compõe a construção Interrogativa_qual.

Depois da atribuição dos ECs, Marção (2018) estabelece as restrições de constituência da construção. Como vimos na seção 3, a constituência tipifica os ECs da construção em termos de outras construções. Assim, podemos assinalar se um EC é composto por uma outra construção já definida no Constructicon. Os ECs da construção Interrogativa_qual possuem essa restrição, o EC Estrutura_argumenta_base é licenciado pela construção Sujeito_predicado e o EC Pronome_qual é licenciado pela construção Pronome_QU, além disso, a restrição Lexeme (lex) especifica que é o pronome "qual" que deve ocupar a posição do pronome, como podemos ver na Figura 8.

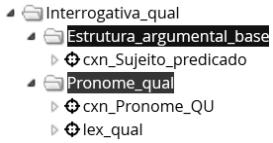


Figura 8. Restrições aplicadas à construção Interrogativa_qual [Marção 2018:85]

Por fim, a construção cadastrada é apresentada no Constructicon na forma da tela da Figura 9.

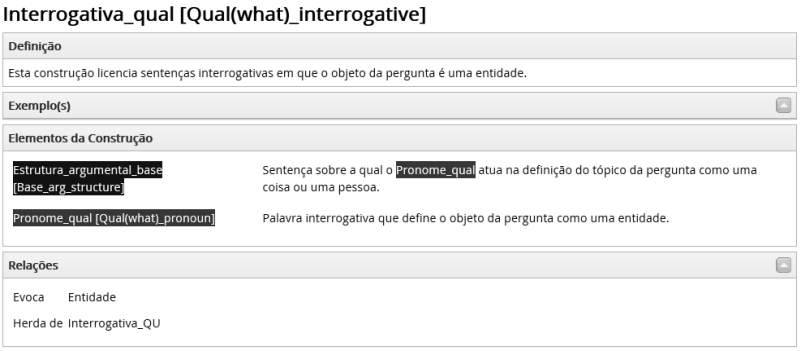


Figura 9. Construção Interrogativa_qual

Na Figura 9 vemos o nome da construção, seguido pelo nome interlingual. Abaixo temos a definição da construção. Em seguida, observamos os ECs Estrutura_argumental_base e Pronome_qual e suas definições. Por fim, vemos que essa construção se relaciona com o Frame Entidade através da relação de *Evocação* e se relaciona a outra construção mais abstrata, Interrogativa_QU, através da relação de *Herança*.

O mesmo procedimento de modelagem se seguiu para as construções Interrogativa_que, Interrogativa_quem, Interrogativa_quando, Interrogativa_quanto, Interrogativa_onde, Interrogativa_como, Interrogativa_QU_preposicionada_razão e Interrogativa_QU_preposicionada_custo, respeitando as especificidades de cada construção [Marção 2018].

A partir da modelagem realizada para o total de onze construções, Marção (2018) propõe a rede construcional destas na Figura 10.

Figura 10. Rede de construções Interrogativas QU modeladas no Constructicon [Marção 2018:101]

A Figura 10 traz o gráfico que mostra as construções modeladas sob as relações de Herança e Evocação. Sendo assim, temos mais acima a construção mais abstrata, a Interrogativa_QU, a qual é herdada pelas demais construções representadas pelos círculos, incluindo a construção Interrogativa_QU_preposicionada, apresentada na parte inferior direita da imagem, a qual também é mais abstrata. Os quadrados representam os frames evocados por essas construções e a relação de Evocação é representada pelas setas que tomam a direção das construções aos Frames.

5. Considerações Finais

Neste texto apresentamos como a modelagem de construções se dá no Constructicon da FN-Br com base nos estudos de Marção (2018), demonstrando como a integração de aspectos da BCG e da Semântica de Frames enriquecem a representação formal e semântica das sentenças. Para isso, foram modeladas onze construções Interrogativas QU-, sendo a construção Interrogativa_QU mais abstrata e herdada pelas construções: Interrogativa_que; Interrogativa_qual; Interrogativa_quem; Interrogativa_quando; Interrogativa_quanto; Interrogativa_onde e Interrogativa_como. Também modelamos a Construção Interrogativa_QU_preposicionada mais abstrata, a qual é herdada pelas construções Interrogativa_QU_preposicionada_razão e Interrogativa_QU_preposicionada_custo.

As construções cuja modelagem foi relatada neste trabalho poderão ser utilizadas em trabalhos futuros que tenham interesse, por exemplo, em analisar os tipos de estruturas argumentais que podem compor as construções Interrogativas QU-, identificar a relevância dessas estruturas para composição das construções Interrogativas QU- e que tipo de relação pode ser estabelecida entre as estruturas argumentais e o elemento QU, entre outras possibilidades de análise.

Referências

Adger, D. (2003). *Core syntax: A minimalist approach* (Vol. 20). Oxford: Oxford University Press.

Augusto, M. R. (2005). QU deslocado e QU in situ no PB: aspectos da derivação lingüística e questões para a aquisição da linguagem. In *IV Congresso Internacional da ABRALIN*. Brasília: ABRALIN, p. 535-542.

Fillmore, C. J. (1985). Syntactic intrusions and the notion of grammatical construction. In *Proceedings of the Annual Meeting of the Berkeley Linguistics Society*. Berkeley, California: UC Berkeley, p. 73-86.

Fillmore, C. J. (2008). Border conflicts: FrameNet meets construction grammar. In *Proceedings of the XIII EURALEX international congress.* Barcelona, Spain: IULA., p. 49-68.

Fillmore, C. J., Lee-Goldman, R., & Rhomieux, R. (2012). The FrameNet Constructicon. In Boas, H. C.; Sag, I. A. *Sign-Based Construction Grammar*. Stanford: CSLI, p. 309-372.

Fillmore, C. J. (2013). Berkeley Construction Grammar. In: Hoffmann, T.; Trousdale, G. (eds.). *The Oxford Handbook of Construction Grammar*. Oxford: Oxford University Press, p. 111-132.

Fried, M., & Östman, J. O. (2004). *Construction Grammar in a cross-language perspective*. Amsterdam: John Benjamins.

Goldberg, A. (2006). *Constructions at Work:* The nature of generalization in language. Oxford: Oxford University Press.

Kay, P., and Fillmore, C. J. (1999). Grammatical constructions and linguistic generalizations: the What's X doing Y? construction. *Language*, 75(1), p. 1-33.

Marção, N. D. (2018). As construções interrogativas QU- no Constructicon da FrameNet Brasil. Dissertação de Mestrado em Linguística – Universidade Federal de Juiz de Fora.

Modesto, M. (2012). O programa minimalista em sua primeira versão. In Figueiredo de Alencar., L.; Othero, G. de A. (eds). *Abordagens computatcionais da Teoria da Gramática*. São Paulo: Mercado de Letras, p. 127-152.

Sag, I. A. (2012). Sign-Based Construction Grammar: An informal synopsis. In Boas, H. C.; Sag, I. A. *Sign-Based Construction Grammar*. Stanford: CSLI, p. 69-202.

Torrent, T. T., Matos, E., Lage, L., Laviola, A., Tavares, T., Almeida, V. G., & Sigiliano, N. (2018). Towards continuity between the lexicon and the constructicon in FrameNet Brasil. In Lyngfelt, B., Borin, L., Ohara, K. & Torrent, T. T. (eds.). *Constructicography: Constructicon development across languages*. Amsterdam: John Benjamins, p. 107-141.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Banco de dados VerboWeb: um panorama do léxico verbal do PB

Márcia Cançado[1], Luana Amaral[1], Letícia Meirelles[1], Thaís Bechir[1], Amanda Oliveira[1]

[1]Faculdade de Letras – Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte – MG – Brasil

mcancado@ufmg.br, luanalopes@let.grad.ufmg.br, lelumeirelles@hotmail.com, thais.carvalhobechir@gmail.com, amanda_noliveira@yahoo.com.br

Abstract. *In this paper, we present the proposal of description and analysis of Brazilian Portuguese verbal data present in the VerboWeb database. Verbs were grouped and classified according to the theoretical-methodological framework of Lexical Semantics, which assumes that some semantic properties of verbs determine their syntactic behavior. We show how the database is structured and the generalizations we can make about the verbal lexicon of our language.*

Resumo. *Neste artigo apresentamos a proposta de descrição e análise de dados verbais do português brasileiro presente no banco de dados VerboWeb. Os verbos foram agrupados e classificados de acordo com o arcabouço teórico-metodológico da Semântica Lexical, que assume que certas propriedades semânticas dos verbos determinam seu comportamento sintático. Mostramos como o banco de dados é estruturado e as generalizações que podemos fazer sobre o léxico verbal de nossa língua.*

Introdução

O banco de dados VerboWeb foi criado pelas professoras doutoras Márcia Cançado, Luana Amaral e pela doutora Letícia Meirelles, no ano de 2017, tendo como ponto de partida, o projeto de descrição verbal do português brasileiro (PB) "Catálogo de Verbos do Português Brasileiro", criado e coordenado pela Profa. Dra. Márcia Cançado, que tem como principal objetivo catalogar os verbos do PB descrevendo suas propriedades sintáticas e semânticas.[1] Atualmente, o desenvolvimento do banco, conta a participação de mais dois membros: a doutoranda Thaís Bechir e a estudante de mestrado Amanda Oliveira.

Neste artigo, apresentamos uma visão geral do banco de dados, mostrando nossa proposta de classificação verbal e metodologia de análise.

1. Estrutura geral do banco de dados

[1] Site do banco: http://www.letras.ufmg.br/verboweb/

A proposta de classificação verbal presente no VerboWeb é norteada por uma linha de pesquisa conhecida como Semântica Lexical ou Interface Sintaxe-Semântica Lexical, que se baseia no pressuposto de que a realização sintática dos argumentos verbais é motivada pela semântica dos verbos [Fillmore 2003 [1970]; Pinker 1989; Levin 1993; Levin; Rappaport Hovav 1995, 2005; Beavers 2010; Wunderlich 2012; Cançado; Godoy; Amaral 2013, 2017; Beavers; Koontz-Garboden 2020].

Atualmente, o banco conta com 1495 verbos que estão distribuídos por 18 classes. A divisão primeira dessas classes é feita de acordo com o aspecto lexical e posteriormente, através de outras propriedades semânticas de significação. Essas propriedades semânticas são representadas em termos de papéis temáticos e através da linguagem de decomposição de predicados. Além da análise semântica, cada classe apresenta propriedades sintáticas, comuns a todos os seus membros, que são decorrentes da significação verbal. Até o presente momento, o banco conta com 43 propriedades sintáticas, exemplificadas por sentenças em cada um dos verbos que as exibem.

O VerboWeb ainda conta com um nível mais específico de classificação, que é o que chamamos de subclasses. Há um total de 7 subclasses no banco, sendo que cada uma delas está relacionada a uma classe específica. Os membros de uma subclasse apresentam as mesmas propriedades sintático-semânticas da classe, mais alguma propriedade semântica específica, que motiva a ocorrência de uma ou mais propriedades sintáticas adicionais.

Por fim, ainda há o que chamamos de propriedades não classificatórias, que são aquelas que não estão relacionadas a uma classe ou subclasse. Elas perpassam por membros de diversas classes, estando relacionadas com apenas parte da significação verbal e até mesmo com elementos pragmáticos. Até o momento, há 8 propriedades não classificatórias no banco.

Todas as classes, subclasses e propriedades são devidamente explicadas e exemplificadas no banco dados de uma maneira interativa. Para consultar essas explicações, basta clicar no conceito desejado, que é aberto um *pop up* com a definição do item em questão e com referências de textos teóricos sobre o assunto.

Nas próximas seções, explicaremos e exemplificaremos esses processos de classificação que utilizamos.

2. Agrupamento pelo aspecto lexical

Como já mencionamos, nosso primeiro nível de classificação baseia-se na noção de aspecto lexical, proposta inicialmente por Vendler (1967) e assumida posteriormente por outros autores, como Comrie (1976) e Smith (1997). De forma geral, o aspecto lexical diz respeito a como as situações descritas pelos verbos desenrolam-se no decorrer do tempo. Esse desenrolar é descrito em termos de traços aspectuais, como dinamicidade, telicidade e duratividade. A partir desses traços, Vendler (1967) propôs a existência de quatro classes aspectuais: atividades, *accomplishments*, *achievements* e estados. Os verbos de atividade descrevem situações dinâmicas, durativas e atélicas, ou seja, que não se encaminham para um resultado, ex: *correr*, *chicotear*, *abraçar*, etc. Verbos de *accomplishmen*t também são dinâmicos e durativos, mas, contrariamente às atividades, são télicos, ou seja, apresentam um resultado final, ex: *machucar*, *acorrentar*, *abastecer*, etc. Verbos de *achievement* são dinâmicos e télicos, como os

accomplishments, porém majoritariamente pontuais, pois focam apenas no resultado final e não da realização de uma ação, ex: *chegar*, *enviuvar*, *adormecer*, etc.[2] Por fim, verbos de estado não são dinâmicos, contrariamente às outras classes, são durativos e atélicos, ex: *amar*, *existir*, *preocupar*, etc.

Até o presente momento, no banco de dados VerboWeb, há 6 classes de verbos de atividade, 7 classes de verbos que denotam *accomplishment*, 2 classes de verbos que apresentam o aspecto lexical de *achievement* e 3 classes que denotam estados.

Tendo mostrado a classificação inicial, que é motivada pelo aspecto lexical, passemos para a descrição das diferentes classes verbais que cada classe aspectual engloba.

2.1 Classes verbais

As classes verbais presentes no VerboWeb são agrupamentos de verbos que apresentam, além do mesmo aspecto lexical, a mesma estrutura argumental, ou seja, o mesmo número e tipo semântico de argumentos, o que faz com que os verbos pertencentes a uma mesma classe tenham os mesmos comportamentos sintáticos. Essa classificação ocorre em um nível intermediário (*medium-grained*) [Levin 2010; Cançado; Amaral 2016; Cançado; Gonçalves 2016] e corresponde, em sua maioria, às classes verbais canônicas propostas na literatura em Semântica Lexical. Nesta seção, mostraremos e exemplificaremos uma classe pertencente a cada aspecto verbal.

Uma classe que faz parte das 6 classes de verbos de atividade é a que denominamos de "verbos de contato mediado pelo corpo". Ela possui 40 verbos do tipo *beijar* e *morder*, que denotam que "X age sobre Y, por meio de um evento mediado pelo corpo" [Cançado; Amaral; Meirelles 2017]. Os verbos pertencentes a essa classe tomam dois argumentos para ter seu sentido saturado (um sujeito e um objeto), que recebem, respectivamente, os papéis temáticos de Agente e o papel que denominamos de Objeto Afetado. Além da estrutura de papéis temáticos, propomos representar o significado dos verbos dessa classe através da seguinte estrutura de decomposição de predicados: [[X ACT$_{<EVENT>}$] ON Y]], na qual a raiz <*EVENT*> corresponde aos eventos *beijo* e *mordida*, enquanto o predicado ON representa a ideia de contato expressa pelos verbos. Além das sentenças- bases (ex: *o menino beijou a menina*; *o cachorro mordeu a menina*), as características semânticas dos verbos licenciam os seguintes comportamentos sintáticos: (i) presença de um sintagma cognato em adjunção (*o menino beijou a menina* ***com um beijo molhado***; *o cachorro mordeu a menina* ***com uma mordida forte***); (ii) formação de nominalização, na qual o sujeito do verbo funciona como complemento nominal (*o beijo do menino na menina*; *a mordida do cachorro na menina*); (iii) realização da passiva eventiva (*a menina foi beijada pelo menino*; *a menina foi mordida pelo cachorro*) e; (iv) fatoração do argumento Objeto Afetado (*o menino beijou a bochecha da menina*/ *o menino beijou a menina na bochecha*; o *cachorro mordeu a perna da menina*/ *o cachorro mordeu a menina na perna*).

[2] Cançado e Amaral (2016), baseadas em Smith (1997), afirmam que certos verbos de *achievement* podem apresentar uma duração, como *amadurecer* e *derreter*, porém continuam denotando eventos que falam apenas do resultado final de uma situação.

Dentre as 7 classes de verbos de *accomplishment*, mostraremos a mais numerosa, que é a classe dos verbos de mudança de estado opcionalmente agentivos, como *quebrar* e *machucar* [Cançado; Godoy; Amaral 2013, 2017]. Fazem parte da classe 464 verbos, que denotam que "X age, causando Y ficar em um determinado estado" [Cançado; Amaral; Meirelles 2017]. Os verbos dessa classe tomam dois argumentos para ter seu sentido completo, sendo que o argumento que ocupa a posição de sujeito pode receber os papéis temáticos de Causa ou de Agente (*a queda quebrou o vaso*/ *o menino machucou o colega intencionalmente*), enquanto o argumento que ocupa a posição de objeto recebe o papel temático de Paciente. A estrutura de decomposição de predicados que representa o sentido da classe é a seguinte: [[X ACT (VOLITION)] CAUSE [BECOME [Y <*RESULT-STATE*>]]]. O predicado VOLITION, entre parênteses, representa a opcionalidade da agentividade na realização do evento, enquanto a associação dos predicados CAUSE e BECOME representam a causação de uma mudança, mais especificamente uma mudança que culmina em um estado resultante, representado pela raiz <*RESULT-STATE*>, correspondente aos estados *quebrado* e *machucado*. As propriedades semânticas da classe licenciam os seguintes comportamentos sintáticos: (i) participação na alternância causativo-incoativa (*a queda quebrou o vaso*/ *o vaso (se) quebrou*; *o acidente machucou a menina*/ *a menina (se) machucou*); (ii) adjunção da causa na forma incoativa (*o vaso (se) quebrou com a queda*; *a menina (se) machucou com o acidente*); (iii) realização das formas passivas eventiva, resultativa e estativa (*o vaso foi/ficou/está quebrado*; *a menina foi/ficou/está machucada*).

Passando para os *achievements*, esse aspecto lexical também apresenta uma classe de mudança de estado, formada por 85 verbos do tipo *amadurecer* e *azedar*. Esses verbos denotam que "X passa a ficar em um determinado estado" [Cançado; Amaral; Meirelles 2017] e apresentam a seguinte estrutura de decomposição de predicados, que é a contraparte da dos verbos de mudança de estado causativos que apresentamos anteriormente: [BECOME [X <*RESULT-STATE*>]]. Essa estrutura evidencia que os verbos da classe falam apenas de um estado final resultante, que corresponde aos estados de *ficar maduro* e *ficar azedo*. Os verbos da classe tomam apenas um argumento para ter seu sentido saturado e esse argumento recebe o papel temático de Paciente. As propriedades semânticas licenciam os seguintes comportamentos sintáticos: (i) participação na alternância incoativa-causativa[3] (*o calor amadureceu a banana*; *o excesso de calor azedou o leite*); (ii) adjunção de causa indireta (*a banana amadureceu com o calor*; *o leite azedou com o excesso de calor*); (iii) realização das formas passivas resultativa e estativa (*a banana ficou/está madura*; *o leite ficou/está azedo*) e; (iv) não aceitação do clítico *se* (**a banana se amadureceu*; **o leite se azedou*).

Por fim, dentre as classes de estado, mostraremos a classe dos verbos de estado psicológico do tipo *amar*. Compõem a classe 34 verbos que denotam que "X está em determinado estado psicológico em relação a algo ou a alguém" [Cançado; Amaral; Meirelles 2017]. Os verbos tomam dois argumentos para ter seu sentido completo, sendo um sujeito Experienciador e um objeto que recebe o papel temático de Objeto Estativo. A estrutura de decomposição de predicados da classe é [X <*PSYCH-STATE*> Y], na qual a categoria ontológica <*PSYCH-STATE*> é subespecificada como um estado

[3] Cançado e Amaral (2016), baseadas em Haspelmath (1993), propõem que os verbos dessa classe são basicamente incoativos, pois não aceitam o clítico *se*, que seria um marcador de alternância no PB.

psicológico que, portanto, relaciona dois indivíduos, no sentido amplo do termo. As propriedades sintáticas derivadas dessas características semânticas são: (i) os verbos licenciam a fatoração do argumento Objeto Estativo (*o rapaz ama o jeito meigo da namorada/ o rapaz ama a namorada pelo seu jeito meigo*; *a moça repudiou as maneiras do rapaz/ A moça repudiou o rapaz por suas maneiras*) e; (ii) licenciam uma perspectiva passiva (*a namorada é amada pelo rapaz*; *o rapaz foi repudiado pela moça*).

Tendo descrito uma classe pertencente a cada uma das classes aspectuais presentes no VerboWeb, descreveremos a seguir uma das subclasses do banco de dados.

2.1.1 Subclasses

O conceito de subclasses está relacionado a um nível mais fino de classificação semântica, conhecido como *fine-grained* [Levin 2010; Cançado; Amaral 2016; Cançado; Gonçalves 2016], pois diz respeito a propriedades que não estão presentes na estrutura argumental dos verbos, mas fazem parte do seu sentido idiossincrático. Segundo Cançado, Amaral e Meirelles (2017, 2018), cada subclasse é relacionada a uma classe verbal no nível *medium-grained* e os verbos que compõem as subclasses, apresentam, além das propriedades sintático-semânticas da classe a qual pertencem, pelo menos uma propriedade sintática a mais, que é decorrente do seu significado específico.

Um exemplo de subclasse é a dos verbos com objeto recíproco. Segundo autores como Dixon (1992), Siloni (2007), Godoy (2008), Bechir (2016), verbos recíprocos são aqueles que exigem que um dos seus argumentos denote dois (ou mais) referentes no mundo, como o verbo *afastar* na sentença *a faxineira afastou os móveis*. No nível *medium-grained*, o verbo *afastar* pertence à classe dos verbos de mudança de estado opcionalmente agentivos, pois apresenta todas as propriedades dos membros da classe: (i) aceita uma Causa ou um Agente como sujeito (*o tremor de terra afastou os móveis*/ *a faxineira afastou os móveis intencionalmente*); (ii) licencia a forma incoativa com o clítico *se* (*os móveis (se) afastaram*); (iii) licencia a causa em adjunção na forma incoativa (*os móveis (se) afastaram com o tremor de terra*) e; (iv) licencia as formas passivas eventiva, resultativa e estativa (*os móveis foram/ficaram/estão afastados*).

Contudo, por ter um argumento que denota dois ou mais referentes (*os móveis*), o verbo *afastar* apresenta os seguintes comportamentos sintáticos específicos: (i) licencia a forma descontínua [Godoy 2008] desse argumento (*a faxineira afastou um móvel do outro*); (ii) licencia a versão descontínua da forma incoativa (*um móvel (se) afastou do outro*) e; (iii) licencia a versão descontínua das formas passivas eventiva, resultativa e estativa (*um móvel foi/ficou/está afastado do outro*). Fazem parte da subclasse de verbos com objeto recíproco 33 verbos como *afastar*, *juntar*, *mesclar*, *separar*, etc.

Até o presente momento, há 7 subclasses no VerboWeb, sendo elas: verbos com objeto recíproco (ex: *afastar*), verbos com sujeito recíproco (ex: *conversar*, *brigar*), verbos de modo de fala (ex: *berrar*, *urrar*); verbos instrumentais com dois instrumentos (ex: *balear*, *flechar*), verbos de remoção (ex: *lavar*, *limpar*), verbos de criação de imagem (ex: *bordar*, *carimbar*) e verbos de contato (*achatar* e *amassar*). Cada uma dessas subclasses está ligada a uma propriedade sintática decorrente do sentido específicos dos verbos que as compõem.

Tendo mostrado e exemplificado o conceito de subclasse que utilizamos no banco, passemos para a explicação daquilo que chamamos de propriedades não classificatórias.

2.2 Propriedades não classificatórias

Além das propriedades sintáticas e semânticas que compõem as classes e subclasses, há alguns comportamentos sintáticos que são decorrentes de propriedades semânticas (e, às vezes, pragmáticas) mais gerais, perpassando por várias classes verbais. Esse tipo de propriedade sintática é chamado de propriedade não classificatória e tem como um de seus exemplos a reflexivização [Camacho 2003 Doron; Rappaport Hovav 2009; Godoy 2012, dentre outros].

A reflexivização está ligada ao conceito de voz reflexiva trazido nas gramáticas tradicionais e basicamente ocorre com verbos transitivos que aceitam um Agente na posição de sujeito. No PB, o pronome reflexivo *se* substitui o argumento em posição de objeto, passando a ter a mesma denotação que o sujeito, como em *a menina se lavou*. Segundo Godoy (2012), o pronome *se* pode ser substituído pela forma *ele(a) mesmo(a)*: *a menina lavou ela mesma*. No banco de dados VerboWeb, 269 dos 1495 verbos analisados realizam a reflexivização, e esses verbos pertencem a várias classes distintas: o *menino se beijou* (verbos de contato mediado pelo corpo); *a menina se ama* (verbos de estado psicológico); *o menino se machucou para não ficar de castigo* (verbos de mudança de estado opcionalmente agentivos).

Há 8 propriedades não classificatórias no VerboWeb até o momento: alternância agente-beneficário (*a menina cortou o cabelo com um ótimo cabeleireiro*), alternância condutor-veículo (*a moça parou a Mercedes na garagem/ a Mercedes parou na garagem*), alternância incoativa periférica (*a roupa já lavou*); alternância parte-todo (*o João quebrou o braço*), reflexiva média (*a menina se enfiou no armário*); reflexiva média da forma descontínua (*uma menina se afastou da outra para dançar*); reflexivização (*a menina se lavou*) e um tipo de alternância aspectual de verbos de *accomplishment* para *estados* com processo de intransitivização (*vidro de azeitona não abre fácil*).

Tendo mostrado as propriedades não classificatórias presentes no banco de dados, finalizaremos a descrição do funcionamento do VerboWeb com algumas breves generalizações sobre o PB.

3 Considerações finais

Como já mencionamos neste artigo, o banco de dados VerboWeb é fruto de um extenso trabalho de pesquisa idealizado e coordenado pela Professora Dra. Márcia Cançado e desenvolvido pelos membros do Núcleo de Pesquisa em Semântica Lexical (NuPeS) da Faculdade de Letras da Universidade Federal de Minas Gerais. Grande parte dos dados e análises que estão presentes no banco é baseada em trabalhos desenvolvidos por membros do grupo desde sua criação até o momento atual.

Ao colocarmos e adaptarmos as análises ao modelo do banco, foi possível perceber algumas generalizações em nossa língua. Por exemplo, 800 verbos do VerboWeb denotam *accomplishments*, sendo consequentemente eventos causativos, enquanto os demais 695 verbos distribuem-se pelas classes de atividade, *achievement* e

estado. Esse padrão reflete a afirmação de que a causação é uma das noções mais básicas presentes em nossa cognição [Langacker 1990; Croft 1991].

Também pudemos notar que a noção de contato é relevante para a classificação verbal. Observamos que os verbos que denotam atividades são divididos em classes de acordo com a presença ou ausência de contato físico entre os indivíduos envolvidos no evento, e no fato de esse contato ser mediado pelo corpo (verbos do tipo *abraçar*), por um instrumento externo ao verbo (verbos do tipo *lavar*) ou por um instrumento incorporado ao nome do verbo (verbos do tipo *chicotear* e *patinar*).

Além disso, ainda pudemos perceber que outras propriedades, como os verbos denotarem uma eventualidade psicológica, expressarem uma relação de posse ou de locação, também são propriedades semânticas que têm impacto na sintaxe, pois temos algumas classes agrupadas a partir dessas propriedades: (i) eventualidade psicológica - verbos de estado psicológico do tipo *amar* e do tipo *preocupar*; (ii) relação de posse - verbos de *locatum* (ex: *acorrentar*), verbos de mudança de estado de posse (ex: *abastecer*) e; (iii) relação de locação - verbos de *location* (ex: *ensacar*), verbos de mudança de estado locativo (ex: *abrigar*), verbos de *achievement* que denotam mudança de lugar (ex: *chegar*) e verbos de estado que expressam a ideia de existência (ex: *existir* e *haver*).

Por fim, gostaríamos de ressaltar que, embora o banco e as análises nele presentes tenham sido desenvolvidas baseadas no arcabouço teórico-metodológico da Semântica Lexical, os dados e as propriedades do VerboWeb podem ser utilizados por pesquisadores de outras linhas de pesquisa que estejam interessados na análise do PB. Além disso, professores dos ensinos fundamental e médio, que desejam aprender mais sobre a semântica e a sintaxe verbal, também podem se valer do banco como uma maneira de facilitar o ensino de nossa língua, já que abordagens linguísticas têm estado cada vez mais presentes nas salas de aula. Por fim, pelo fato de fornecer um extenso panorama do funcionamento do léxico verbal do PB, o banco talvez possa ser útil para a checagem de dados de tradução, já que apresenta uma série de possibilidades de ocorrências sintáticas de nossa língua.

Referências

Beavers, J. (2010). The Structure of Lexical Meaning: Why Semantics Really Matters. *Language*. 86, p. 821-864.

Beavers, J.; Koontz-Garboden, A. (2020), The Roots of Verbal Meaning. Oxford: Oxford University Press.

Bechir, T. (2016). *Os verbos recíprocos intransitivos no português brasileiro*. Monografia de Bacharelado em Português (Estudos Linguísticos). Faculdade de Letras, UFMG, Belo Horizonte, Brasil.

Cançado, M., & Amaral, L. (2016). *Introdução à Semântica Lexical*: *Papéis Temáticos,aspecto lexical e decomposição de predicados*. Editora Vozes, Petrópolis, RJ, Brasil.

Cançado, M, Amaral, L., & Meirelles, L. (2017). *VerboWeb: classificação sintático-semântica dos verbos do português brasileiro*. Banco de dados lexicais. UFMG. Disponível em: http://www.letras.ufmg.br/verboweb.

Cançado, M., Amaral, L., & Meirelles, L. (2018). VerboWeb: uma proposta de classificação verbal. *Revista da Anpoll*, n. 46, v. 1.

Cançado, M., Godoy, L., & Amaral, L. (2013). *Catálogo de verbos do português brasileiro: classificação verbal segundo a decomposição de predicados*. Vol I. Verbos de mudança, 1 ed. Editora UFMG, Belo Horizonte, Brasil.

Cançado, M., Godoy, L., & Amaral, L. (2017). *Catálogo de verbos do português brasileiro: classificação verbal segundo a decomposição de predicados*. Vol I. Verbos de mudança, 2 ed. Edição Revisada Amazon. 2017.

Cançado, M., & Gonçalves, A. (2016). Lexical Semantics: verb classes and alternations. In L. Wetzels, S. Menuzzi, & J. Costa (Eds.), *The Handbook of Portuguese Linguistics*. Willey/Blackwell, 374-391.

Camacho, R. (2003). Em defesa da categoria de voz média no português. *DELTA*, v. 19, n.1, p. 91-122.

Comrie, B. (1976), Aspect: an Introduction to the study of verbal aspect and related problems. Cambridge: Cambridge University Press.

Croft, W. (1991), Syntactic Categories and Grammatical Relations. Chicago: University of Chicago Press.

Dixon, R. (1992), A new approach to English grammar, on semantic principles. Oxford: Clarendon Press.

Doron, E.; Rapapport Hovav, M. (2009) "A unified approach to reflexivization in Semitic and Romance". In: Bendjaballah et al. (orgs.) Brill's Annual of Afroasiatic Languages and Linguistics 1. Leiden: Brill, p. 75–105.

Godoy, L. (2008), Os verbos recíprocos no PB: interface sintaxe-semântica lexical. (Mestrado em Estudos Linguísticos) – Faculdade de Letras, Universidade Federal de Minas Gerais, Brasil.

Godoy, L. (2012), A reflexivização no português brasileiro e a decomposição semântica de predicados. Tese (Doutorado em Estudos Linguísticos) – Faculdade de Letras, UFMG, Belo Horizonte.

Fillmore, C. (2003 [1970]) "The grammar of hitting and breaking". In: Fillmore (ed.), Form and meaning in language: Papers on semantic roles, p. 123–139. Stanford: CSLI Publications

Haspelmath, M. (1993) "More on typology of inchoative/causative verb alternations". In: Comrie & Polinsky. Causatives and transitivity. Amsterdam: John Benjamins, p. 87-120.

Langacker, R. (1990), Concept, Image and Symbol: The Cognitive Basis of Grammar. Berlin: Mouton de Gruyter.

Levin, B. (1993), English verb classes and alternations: A preliminary investigation. Chicago: University of Chicago Press.

Levin, B. (2010) "What is the best grain- size for defining verb classes?" Conference on Word Classes: nature, Typology, Computational Representations, Second TRiPLE international Conference, Università Roma Tre, Rome, March 24–26.

Levin, B.; Rappaport Hovav, M. (1995), Unaccusativity: At the syntax-lexical semantics interface. Cambridge: The MIT Press.

Levin, B.; Rappaport Hovav, M. (2005), Argument realization. Cambridge: Cambridge University Press.

Pinker, S. (1989), Learnability and Cognition: The Acquisition of Argument Structure. Cambridge, MA: MIT Press.

Siloni, T. (2007) “The syntax of reciprocal verbs: an overview”. In: König and Gast (eds.) Reciprocals and reflexives: cross-linguistics and theoretical explorations. Berlin: Mouton de Gruyter.

Smith, C. (1997), The Parameter of Aspect. Dordrecht: Kluwer.

Vendler, Z. (1967), Linguistics in philosophy. Ithaca: Cornell.

Wunderlich, D. (2012) “Lexical Decomposition in Grammar”. In: Werning, Hinzen and Machery (Eds.). The Oxford Handbook of Compositionality. Oxford: Oxford University Press, p. 307-327.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Engenharia de *features* linguísticas para classificação de triplas relacionais

Elian Conceição Luz[1,2], **Camilla Rastely da Silva**[2], **Daniela Barreiro Claro**[1]

[1] FORMAS Research Group, Departamento de Ciência da Computação
Instituto de Computação – Universidade Federal da Bahia,
40170-110 – Salvador, BA, Brazil

[2]Instituto de Letras - Universidade Federal da Bahia
40170-110 – Salvador, BA, Brazil

{elianc,camillars,dclaro}@ufba.br

Abstract. *In this study, linguistic features were listed for classification of triples, based on a parallel corpus in Galician, Brazilian Portuguese (PT-BR) and European Spanish (EE). In the experiments, at the syntactic level, a relevant performance of features that offer greater difficulty to extract valid triples was observed, such as the co-related null object/subject and the verb-subject inversion, as well as relational triples that form ungrammatical sub-sentences. At the morphological level, it was observed that the grammatical class of the initial word of each sub-sentence, especially when they are prepositions, were relevant for the classification of the triples.*

Resumo. *Neste estudo, elencaram-se features morfossintáticas para classificação de triplas, com base em um corpus paralelo em Galego, Português do Brasil e Espanhol Europeu. Nos experimentos, a nível sintático, observou-se desempenho relevante de features que oferecem maior dificuldade para extrair triplas válidas, como as co-relacionadas a objeto/sujeito nulo e a inversão verbo-sujeito, bem como triplas relacionais que formam sub-sentenças agramaticais. A nível morfológico, observou-se que a classe gramatical do vocábulo inicial de cada sub-sentença, sobretudo quando são preposições, foram relevantes para a classificação das triplas.*

1. Introdução

O exponencial crescimento do volume de documentos digitais escritos em linguagem natural impulsiona a demanda por modelos automáticos capazes de extrair informação de dados não-estruturados. Nessa perspectiva, a Extração de Informação Aberta (EIA) possibilita a estruturação da linguagem natural em triplas relacionais ao processar as sentenças de um texto, obtendo uma estrutura composta por três elementos (arg1, relação, arg2) [Barbosa 2018]. Por exemplo, a partir da sentença 'João ama Maria', pode-se extrair a tripla válida ('João', 'ama', 'Maria').

A motivação de compor tarefas de PLN em perspectiva multilíngua é ampliar o alcance dos métodos de EIA para fora do alcance da língua inglesa. Em relação ao Português (PT-BR) e ao Espanhol Europeu (EE), por exemplo, mesmo sendo línguas de

ampla circulação e com números expressivos de falantes, ambas apresentam recursos escassos para automação de tarefas do PLN, o que é ainda mais evidente em relação à língua galega. Nessa perspetiva, este estudo favorece o desenvolvimento de métodos capazes de extrair informação de textos redigidos nessas três línguas. Outrossim, a descrição de características de língua propicia maior independência de domínio, contribuindo para a melhoria da Extração de Informação Aberta (EIA) e para a capacitação linguística dessas línguas [Calvet 2005][Gauger 1989].

Tradicionalmente, a EIA inclui métodos de extração de triplas que, em sua maioria, foram desenvolvidos com base em corpus de língua inglesa, considerando suas especificidades linguísticas, como a maior predominância do preenchimento do sujeito e a ordem sujeito-verbo, características que não são predominantes em línguas românicas[Barbosa 2018]. Muitos estudos propuseram a tradução das *features* para a língua-alvo, o que não permitiu obter resultados satisfatórios. A dificuldade aumenta quando se trata de abordagens multilínguas. Assim, o principal propósito deste trabalho foi determinar as *features* linguísticas por meio das triplas extraídas de corpora em línguas românicas: Galego, Português (PT-BR) e Espanhol Europeu (EE). A metodologia adotada analisou as características linguísticas genéricas por meio de uma revisão de estudos da Linguística Formal, em destaque para os estudos de base gerativista [Chomsky and Lasnik 2008] [Chomsky 2014].

Este artigo está estruturado como segue: a seção 2 apresenta a problemática do estudo e hipóteses para o direcionamento dos experimentos; a seção 3 descreve as *features* elencadas para o desenvolvimento dos experimentos, construídas com base na descrição linguística das línguas-alvo; na seção 4, destacam-se a constituição do corpus, os experimentos e a avaliação das *features*; e , por fim, a seção 5 consolida os principais resultados e finaliza o estudo realizado.

2. *Features* linguísticas nas três línguas românicas

Tradicionalmente, a Extração de Informação em documentos busca explorar padrões que expressam relações predefinidas, como geral-específico, parte-todo, localidade e pertencimento. Em sintonia com o ambiente multifacetado da *Web*, a Extração de Informação Aberta (EIA) não limita o tipo de relação, possibilitando a análise de texto em diferentes domínios. Uma possibilidade é a extração de triplas relacionais por meio de métodos de Processamento de Linguagem Natural (PLN), cuja estrutura composta por três elementos (*arg1*, *relação*, *arg2*) é capaz de representar conhecimento de forma mais flexível, a exemplo de atributos, eventos, fatos e entidades [Barbosa 2018].

No entanto, em sua maioria, os métodos de EIA ainda estão focalizados em domínios linguísticos específicos, enquanto cada vez mais a Web concentra textos de diversas línguas e domínios, evidenciando as especificidades dos métodos que dificilmente são replicáveis a contextos mais amplos. Ao considerar as línguas selecionadas para estudo, observam-se aproximações e divergências com possíveis implicações para extração e validação de triplas relacionais em perspectiva multilíngua. Para tal, faz-se necessária descrição linguística dos domínios de língua para a assertiva execução de tarefas de extração e validação de triplas relacionais. Como um exercício da dificuldade em elencar convergências entre esses domínios, pode-se tomar como exemplo as sentenças e suas respectivas extrações apresentadas logo abaixo:

Galego

Bastas querer atopar os teus funcionarios, que [*sujeito nulo*] aparecen no mesmo instante.
([sujeito nulo], bastas querer atopar, os teus funcionarios)

Português Brasileiro

Basta você querer encontrar seus funcionários, que eles aparecem no mesmo instante.[Fanjul 2014]
(você, basta querer encontrar, seus funcionários)

Espanhol Europeu

Basta que quieras encontrar a tus empleados, que [*sujeito nulo*] aparecen en el mismo instante. [Fanjul 2014]
([sujeito nulo], basta que quieras encontrar a, tus empleados).

Em estudos anteriores, demonstrou-se a importância das preposições para análise da estrutura das sentenças. Observou-se que sentenças maiores apresentam uma maior dificuldade para as tarefas de extração, pois elas, geralmente, formam períodos mais complexos. Pode-se, também, considerar que a presença de determinada classe gramatical é mais frequente em cada elemento da tripla, como são os nomes no argumento 1 e no argumento 2, por outro lado, os verbos flexionados são necessários na relacional entre os argumentos.

Por fim, as classes gramaticais das palavras em posição final e inicial indicam características importantes, como pode ser observado na sétima *feature* do Reverb [Fader et al. 2011]. Por exemplo, o conjunto de *features* proposto no modelo indica a transitividade verbal: se é direta ou indireta. Outrossim, pode ser indicativa de inversão da ordem frasal, bem como deslocamento com formação de tópico, com o deslocamento do complemento adverbial de lugar. Neste caso, é possível identificar o deslocamento pela posição da preposição 'em' no início da sentença.

3. *Engenharia de Features*

A partir da revisão bibliográfica de *features* propostas de trabalhos relacionados, selecionaram-se algumas características possivelmente relevantes para classificação de triplas relacionais no Galego, Português Brasileiro e Espanhol Europeu. Fez-se necessário conhecimento de domínio para análise assertiva das características de língua e correta aplicação das técnicas de análise de dados. O objetivo foi extrair o máximo de informações relevantes para análise, o que foi consolidado nas *features* elencadas na Tabela 1.

Ao observar as *features*, destaca-se, por exemplo, que as preposições ajudam a identificar a função sintática dos itens de uma sentença.

Exemplo 1

(1) *Na segunda, João foi à escola*.

A presença de tópico está relacionada a outros fenômenos sintáticos, como o sujeito nulo [de Castilho et al. 1973] [Chomsky and Lasnik 2008].

Exemplo 2

Table 1. *Features* propostas neste estudo

N	*Feature*
1	O Arg1 não é vazio e está contido na sentença
2	O Arg2 não é vazio e está contido na sentença
3	A Rel está contida na sentença
4	Classe gramatical da palavra no início da sentença
5	Classe gramatical da palavra no início do Arg1
6	Classe gramatical da palavra no início do Arg2
7	Classe gramatical da palavra no início do Rel
8	As preposições em início de sentença
9	As preposições em início de Arg1
10	As preposições em início de Arg2

(2) Onde está João?

(3) No domingo, [*sujeito nulo*] vai para escola.

No corpus estudado, é possível observar esse mesmo fenômeno em uma das sentenças nas três línguas estudadas. A partir dessa amostra, é possível observar um exemplo de como o deslocamento se relaciona com a ocorrência do sujeito nulo nessas línguas.

Excerto do corpus 1

(4) **GL**

Sentença:Na política interna [*sujeito nulo*] mostrouse sempre a favor dun goberno de conciliación nacional e foi contrario á unificación de Moldávia coa Romanía.

Tripla válida: ([*sujeito nulo*], foi contrario á, unificación de Moldávia coa Romanía)

(5) **EE**

Sentença: En política interior [*sujeito nulo*] se mostró siempre a favor de un gobierno de conciliación nacional y fue contrario a la unificación de Moldavia con Romanía.

Tripla válida: ([*sujeito nulo*], fue contrario a, la unificación de Moldavia con Romanía)

(6) **PT-BR**

Sentença: Na politica interna, [*sujeito nulo*] mostrou-se sempre a favor de um governo de conciliação nacional e foi contrário à unificação da Moldávia com a România.

Tripla válida: ([*sujeito nulo*], foi contrário à, unificação da Moldávia com a România)

Nesse ponto, observa-se que há uma relação entre o deslocamento, que pode ser identificado com as *features* 4 e 8. Posto que a presença de um determinante ou nome em início de sentença indica que não houve deslocamento, enquanto a presença da preposição indica que há esse deslocamento, o que está correlacionado com a aparição do sujeito

nulo, que se manifesta no esvaziamento do Arg 1, que pode ser identificado na *feature* 1 [de Castilho et al. 1973] [Chomsky and Lasnik 2008].

O fato dos argumentos (*Arg1* e *Arg2*) e a relação (*Rel*) estarem ou não na sentença apresenta indício do processamento utilizado para a extração de triplas, principalmente por meio dos métodos baseados em regras. Enquanto os casos em que o Arg2 estavam vazios apresentam-se como inválidos em sua totalidade, o fato de a relação (*Rel*) ter sido alterada na tripla indica um processo mais complexo de extração.

Excerto do corpus 2

(7) **GL**

Sentença: Fonte:cronista cumpre do diego video diego el 10 Maradona prepara a lista para enfrentar a España. Coa clasificación ao Mundial, o entrenador Diego Maradona dará a coñecer este venres a lista de convocados para o partido ante España, o 14 de novembre en Madrid.

Tripla inválida: (el 10 Maradona, prepara para, enfrentar a España Coa)

Tripla válida: (el 10 Maradona, prepara, a lista)

(8) **EE**

Sentença: Fuente: cronista cumple del diego video diego el 10 Maradona prepara la lista para enfrentar a España. Con la clasificación al Mundial, el entrenador Diego Maradona dará a conocer este viernes la lista de convocados para el partido ante España, el 14 de noviembre en Madrid.

Tripla inválida: (el 10 Maradona, prepara para, enfrentar a España Con)

Tripla válida: (el entrenador Diego Maradona, dará, la lista)

(9) **PT-BR**

Sentença: Fonte: repórter encontra vídeo de Diego, o 10 Maradona, Diego prepara a lista para enfrentar a Espanha. Com a classificação para a Copa do Mundo, o técnico Diego Maradona anunciará nesta sexta-feira a lista de convocações para a partida contra a Espanha, no dia 14 de novembro, em Madri.

Tripla inválida: (o 10 Maradona, prepara para, enfrentar a Espanha Com)

Tripla válida: (o técnico Diego Maradona, anunciará, a lista)

Por fim, observamos que para cada elemento da tripla, esperam-se determinadas classes gramaticais em posição inicial, sobretudo, no *Arg 1* e na *Rel*, nas quais, respectivamente, são mais frequentes nomes ou determinantes e verbos flexionados.

Excerto do corpus 3:

GL

(10) **Sentença**: Cable audio/video estándar para Xbox 360: Conecta instantáneamente aos xogadores ao mundo de Xbox 360, introduciéndolles en gráficos e xogos de última xeneración utilizando conexión é de definición estándar.

Tripla inválida: (xogos de última xeneración, utilizando, conexión)

EE

(11) **Sentença**: Cable audio/video estándar para Xbox 360: Conecta instantáneamente a los jugadores al mundo de Xbox 360, introduciéndoles en gráficos y juegos de última generación utilizando conexión es de definición estándar.

Tripla inválida: (juegos de última generación, utilizando, conexión)

PT-BR

(12) **Sentença**: Cabo de áudio / vídeo padrão do Xbox 360: Conecta instantaneamente os jogadores ao mundo do Xbox 360, apresentando-os a gráficos e jogos de ponta usando conexão de definição padrão.

Tripla inválida: (jogos de ponta, usando, conexão).

Destaca-se que por outro lado, no *Arg 2*, há uma maior variedade, posto que além dos nomes e determinantes, também são frequentes preposições e outras classes gramaticais. No entanto, notou-se que a presença, por exemplo, de verbos na posição inicial são mais frequentes em triplas inválidas, como pode ser visto no excerto 03: ('o 10 Maradona', 'para enfrentar a Espanha', 'prepara a lista de convocações'). Um ponto de dificuldade dessa extração é justamente a ordem que cada elemento da tripla se apresenta na sentença, mas novamente, a posição inicial da classe gramatical ou da preposição que inicia a *Rel* e o *Arg 2* são pertinentes.

4. Experimentos e resultados

Essa seção descreve o corpus paralelo criado e os experimentos para validar o conjunto de *features* definida para as três línguas.

4.1. Constituição do corpus paralelo em Galego, Português Brasileiro e Espanhol Europeu

Um ponto sensível do desenvolvimento do método proposto é a constituição de *corpora* em Galego, Português (PT-BR) e Espanhol Europeu (EE). A criação de um corpus paralelo permitiu analisar as *features* genéricas. Para tanto, selecionou-se uma base em Espanhol que já fora submetida a experimentos por outros pesquisadores [Barbosa 2018] [Gamallo and Garcia 2011]. O corpus foi criado com 371 triplas em Espanhol Europeu (EE), das quais 271 inválidas e 100 válidas acompanhadas das sentenças das quais foram extraídas. O processo de tradução para o Português (PT-BR) e para o Galego foi realizado por especialistas em Português (PT-BR) e em Galego.

4.2. Experimentos

Antes de realizar os experimentos com os algoritmos de classificação, em pré-processamento, realizaram-se duas tarefas de PLN, a segmentação de sentença em palavras (tokenizer) e classificação morfológica (POS taggers). Logo após esses procedimentos, realizou-se a transformação das variáveis categóricas em binárias, na qual, por exemplo, a *feature* "classe gramatical da palavra no início da sentença" foi substituída por variáveis binárias que indicam a presença (1) ou não (0) de uma determinada classe gramatical na posição inicial das sentenças.

Três experimentos foram realizados com os seguintes modelos de classificação: Regressão Logística, *Lazy Learning* e Árvore de decisão a fim de comparar o desempenho

de cada um na classificação das triplas em válidas ou inválidas. Para o *Lazy Learning*, antes de sua execução, executou-se um algoritmo de busca para localizar a quantidade ideal de vizinhos próximos com base na melhor precisão encontrada.

A seleção de *features* ocorreu de forma iterativa. Ao observar *features* que tratavam de um mesmo aspecto linguístico ou que não colaboram para a melhoria do modelo da classificação, realizaram-se ajustes, mantendo as que apresentavam um melhor desempenho na avaliação, sendo o resultado final apresentado na Tabela 1.

4.3. Avaliação

Com o intuito de avaliar as *features* e averiguar o desempenho das *features*, o corpus foi dividindo em holdlout de 70% para treino e %30 para teste. Além disso, para evitar overfitting, o modelo k-flod 10 foi utilizado. Ao tomar como base estudos realizados por outros pesquisadores, adotou-se a precisão como a medida mais adequada para quantificar os resultados das *features* como relevantes para a EIA. Em ordem de prioridade, o estudo considerou precisão, f1, *recall* e acurácia.

Table 2. *Corpus* paralelo - Galego

Modelo	Precisão	F1	Revocação	Acurácia
Regressão Logística	**0.8296549**	**0.8625751**	0.9029703	0.7650192
Árvore de Decisão	0.7658444	0.8478815	**0.9089476**	**0.7658444**
37 vizinhos próximos	0.7877068	0.8267316	0.8788258	0.7386713

Table 3. *Corpus* paralelo - Português do Brasil

Modelo	Precisão	F1	Revocação	Acurácia
Regressão Logística	0.7727757	**0.8334183**	0.913047	**0.728790**
Árvore de decisão	**0.7754385**	0.8297937	0.9042388	0.7270572
17 vizinhos próximos	0.7445605	0.8222542	**0.9287656**	0.7169777

Table 4. *Corpus* paralelo - Espanhol

Modelo	Precisão	F1	Revocação	Acurácia
Árvore de decisão	0.773257	**0.852036**	**0.9421285**	0.7185568
Regressão Logística	**0.8142896**	0.8193759	0.8368115	0.7210765
57 vizinhos próximos	0.781657	0.8235130	0.8989786	**0.7472554**

A base em Espanhol Europeu foi, também, submetida a teste em experimento realizado por [Barbosa 2018], contudo com um pré-processamento distinto. Nesse experimento, os resultados obtidos pelo modelo Regressão Logística pelas *features* propostas pelo CMULTI foram precisão 0.717, f1 0.828, revocação 0.982 e acurácia 0.714 e pelo ReVerb foram precisão 0.709, f1 0.818, revocação 0.968 e acurácia 0.698. Assim, com excesso da revocação, as demais medidas de avaliação apontam para um melhor desempenho na classificação.

5. Conclusão

Neste trabalho, apresentou-se uma pesquisa dedicada a levantar o maior número de *features* para classificação de triplas relacionais de *corpora* em Galego, Português Brasileiro e Espanhol Europeu. Nesse ponto, destaca-se, também, que aspectos linguísticos comuns ao Galego, Português e Espanhol podem impor maior dificuldade aos modelos de extração arquitetados para realidade da língua inglesa, pois a estrutura Sujeito- Verbo-Objeto (SVO) é menos frequentes nessas línguas.

Assim, as *features* selecionadas nesse estudo possibilitaram a análise das triplas relacionais em todos os conjuntos testados. Entre as *features*, destacaram as relacionadas à presença das preposições e demais classes gramaticas em posição inicial. Dessa forma, os experimentos realizados relacionam estudos da Linguística Formal à Engenharia de *Features*, colaborando para Extração de Informação Aberta em perspectiva multilíngua. Em novos experimentos, os estudo podem ser enriquecidos com a análise de dependência, adoção de novos modelos de classificação e um estudo aprofundado sobre as línguas abordadas.

References

Barbosa, G. C. G. (2018). Utilizando *Features* multi-idioma para classificação de triplas relacionais em português, inglês e espanhol. Masters thesis, Universidade Federal da Bahia, Salvador.

Calvet, L. (2005). *As Politicas Linguísticas*. Parabola, São Paulo.

Chomsky, N. (2014). *The minimalist program*. MIT press, Cambridge.

Chomsky, N. and Lasnik, H. (2008). The theory of principles and parameters. In *Syntax*, pages 506–569. De Gruyter Mouton.

de Castilho, A. T., Kato, M. A., and do Nascimento, M., editors (1973). *Gramática do Português Culto falado no Brasil: a construção da sentença*. Fondo de Cultura Económica, Cidade do México.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 8(4):2011. p. 1535–1545.

Fanjul, A. P. (2014). Conhecendo assimetrias: a ocorrência de pronomes pessoais. In *Syntax*, pages 29–50. Parabola editorial.

Gamallo, P. and Garcia, M. (2011). Multilingual open information extraction. *Portuguese Conference on Artificial Intelligence.*, 8(4):p. 1535–1545.

Gauger, H.-M. (1989). *Introductión a la lingüistica románica*. Gredos, Madrid.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Descrição de uma metodologia desenvolvida para revisão de um léxico de palavras de emoção

Barbara C. Ramos

PPGEL, PUC-Rio
Rua Marquês de São Vicente, 225 Rio de Janeiro, Brasil
barbaracmpramos@gmail.com

Abstract. *This article describes a methodology that was developed to review Emocionário, a lexicon of emotion words in Portugese from Linguateca's AC/DC project. This first portion of the review was carried out in seven of the 24 groups of Emocionário, being them "Desespero", "Esperança", "Humildade", "Pena", "Satisfação", "Saudade" and "Surpresa". The methodology is then illustrated, making use of examples taken from the corpus itself for each of the steps. Some of the main changes are documented and discussed at the end.*

Resumo. *Este artigo tem por objetivo descrever a metodologia desenvolvida para revisar o Emocionário, léxico de palavras de emoção do projeto AC/DC, da Linguateca. Esta primeira parcela da revisão foi realizada em sete dos 24 grupos do Emocionário, sendo eles "Desespero", "Esperança", "Humildade", "Pena", "Satisfação", "Saudade" e "Surpresa". A metodologia é descrita de forma detalhada, fazendo uso de exemplos retirados do próprio corpus para ilustrar cada passo. Ao final, algumas das principais mudanças realizadas são documentadas e discutidas.*

1. Introdução

O léxico de emoções do AC/DC (Santos e Bick, 2000) foi desenvolvido no âmbito da Gramateca inicialmente por Cristina Mota e Diana Santos (2015). Sua primeira fase de desenvolvimento teve o objetivo de aumentar a informação semântica nos corpora da Linguateca. A princípio, foi construído a partir de dicionários e ontologias lexicais e revisado manualmente (Mota e Santos, 2015). A construção do léxico, como ressaltado pelas próprias autoras, foi um pontapé inicial para a documentação e anotação da menção de emoção que precisaria de revisão e reavaliação. Nessa primeira fase do trabalho, uma grande quantidade de informação foi agrupada e, também, diversos problemas identificados. O objetivo principal, que foi cumprido, era oferecer um conjunto de dados à comunidade linguística como uma tentativa de anotar emoções. A segunda fase do projeto (Santos et al., 2021) extinguiu as subdivisões da anotação das palavras de emoção em "emomin" e "emomax", que marcavam lemas com pelo menos um sentido de emoção e lemas com a maior parte das ocorrências no corpus indicando emoção, respectivamente. Essa fase também criou quatro novos grupos, totalizando 24

grupos de emoção[1] que comportavam aproximadamente 2.800 lemas diferentes e cerca de 8.000 formas de palavras que são usadas na língua portuguesa para descrever emoção.

A partir de 2019, o projeto de emoções do AC/DC passou a se chamar Emocionário e entrou em sua terceira fase, que consiste em reavaliar, revisar e reorganizar o léxico de emoções que já existia, criando um material de anotação ainda mais refinado e preciso. A partir das regras baseadas no Emocionário, é feita a anotação nos corpora. Essas regras podem ser mais descomplicadas, indicando que a palavra é uma emoção, ou mais complexas – que acabam ocorrendo em maioria – quando a palavra indica emoção em certos contextos. O léxico do Emocionário tem como principais características (i) a possibilidade de sobreposição de grupos, podendo um mesmo lema fazer parte de mais de um grupo de emoção (como "Felicidade" e "Satisfação"), trabalhando de forma complementar segundo os sentidos das palavras; (ii) a abrangência semântica de cada grupo, que faz com que lemas como *agitado*, *exaltado* e *furioso* pertençam ao mesmo grupo, mesmo estando distantes uns dos outros considerando "níveis de fúria"; e (iii) a inclusão de palavras independentemente de sua frequência no corpus, sinalizando que carregam emoção em algum contexto. O Emocionário é, portanto, o projeto de anotação semântica das emoções, em um sentido mais amplo, e o léxico das emoções, em um sentido mais estrito, que é o resultado final observado pelo usuário da Linguateca.

Uma primeira parcela da revisão foi realizada em 2020, em sete dos 24 grupos que compõem o léxico ("Desespero", "Esperança", "Humildade", "Pena", "Satisfação", "Saudade" e "Surpresa"). Um dos principais objetivos era mensurar o desafio de analisar aspectos semânticos de uma língua e tomar consciência das características da análise de emoção, alternando leituras distantes e aproximadas, para aprimorar a anotação semântica do léxico e auxiliar a criação de subsídios para a anotação semiautomática do campo semântico das emoções na língua.

Nas seções 2 a 5, a metodologia é descrita detalhadamente, fazendo uso de exemplos retirados do próprio corpus no processo de aplicação do método para ilustrar os passos. Em seguida, são discutidas algumas mudanças realizadas nos grupos já revisados e apresentados comentários e resultados quantitativos.

2. Método de Revisão do Emocionário

A revisão dos grupos envolve recursos da linguística com corpus como linhas de concordância, listas de frequência, distribuição por campo semântico e lemas; alterna análises de dados nas formas quantitativa e qualitativa através de leitura distante e leitura aproximada; e conta com consulta em dicionários e ferramentas de pesquisa na internet. O primeiro movimento é a análise individual dos lemas, o passo 1, que consiste em buscar se eles já estão anotados semanticamente como emoção e, se sim, como parte do grupo em questão. Na Imagem 1 está o exemplo de *desesperador* cujas entradas em

[1] Os 24 grupos da segunda fase do léxico eram "Admirar", "Alívio", "Amor", "Coragem", "Desejo", "Desespero", "Esperança", "Felicidade", "Fúria", "Genérica, "Gratidão", "Humildade", "Infelicidade", "Ingratidão", "Insatisfação", "Inveja", "Medo", "Ódio", "Orgulho", "Pena", "Satisfação", "Saudade", "Surpresa" e "Vergonha".

corpus já são parte do grupo “Desespero”. Neste caso, o processo consiste apenas em conferir as linhas de concordância e confirmar se os usos do lema condizem com a anotação para mantê-lo no grupo.

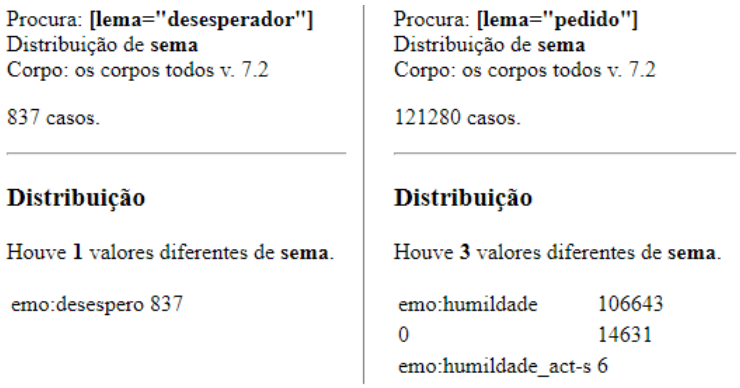


Imagem 1. Exemplo do Passo 1 com a distribuição semântica dos lemas *desesperador* e *pedido*

Para os demais casos, que são a maioria nos grupos do Emocionário já revisados, considerando a imensa variedade semântica do português, a metodologia segue um total de sete passos que podem ser agrupados em três momentos. Esses passos são apresentados em detalhe nas seções 3, 4 e 5. No primeiro momento, realizado o passo 1, seguem-se os passos 2 e 3 que observam a contextualização dos lemas não anotados e anotados semanticamente, com ou sem emoção, visitando as linhas de concordância em que aparecem. Em um segundo momento, os passos 4 e 5 confirmam ou buscam por informações adicionais sobre os sentidos dos lemas sob análise. No terceiro momento, os passos 6 e 7 concluem o processo, que acontece a partir da identificação de novos lemas que podem ser adicionados nos grupos do léxico e da definição dos lemas que serão mantidos, retirados ou mesmo transferidos ou adicionados a outro grupo do Emocionário.

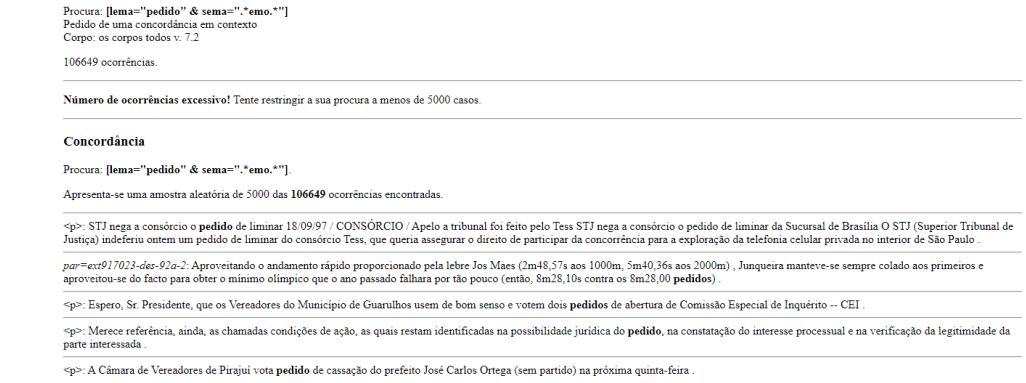
3. Análise da Anotação Semântica no Contexto dos Corpora

No primeiro momento, após selecionar o lema para análise, busco quais etiquetas semânticas ele carrega (pela opção “distribuição por anotação semântica” no AC/DC), concretizando o passo 1 da metodologia, como ilustrado na seção 2 pelo lema *desesperador*. A Imagem 1 mostra também resultados referentes ao lema *pedido*, que integra o grupo “Humildade”.

No passo 2, busco pelas linhas de concordância das entradas – ainda referentes a *pedido* – que estão anotadas como emoção (sema=”emo”), como mostra a expressão de busca[2] na Imagem 2, para confirmar se a anotação condiz com os contextos de uso do lema. A partir desses exemplos, fica claro que o uso do lema *pedido* não tem sentido emocional em particular.

[2]A expressão de busca correspondente é [lema="pedido" & sema=".*emo.*"]

No passo 3, busco exemplos por linhas de concordância das entradas que estão anotadas com outros campos semânticos ou estão sem anotação semântica (sema="0" e sema!="emo") para confirmar os usos nos recortes apresentados pelo AC/DC. Nos casos em que há muitas ocorrências, a ferramenta apresenta uma listagem randômica e, a partir da amostragem automática, seleciono as linhas para leitura, também aleatoriamente.


Procura: **[lema="pedido" & sema=".*emo.*"]**
Pedido de uma concordância em contexto
Corpo: os corpos todos v. 7.2

106649 ocorrências.

Número de ocorrências excessivo! Tente restringir a sua procura a menos de 5000 casos.

Concordância

Procura: **[lema="pedido" & sema=".*emo.*"]**.

Apresenta-se uma amostra aleatória de 5000 das **106649** ocorrências encontradas.

<p>: STJ nega a consórcio o **pedido** de liminar 18/09/97 / CONSÓRCIO / Apelo a tribunal foi feito pelo Tess STJ nega a consórcio o pedido de liminar da Sucursal de Brasília O STJ (Superior Tribunal de Justiça) indeferiu ontem um pedido de liminar do consórcio Tess, que queria assegurar o direito de participar da concorrência para a exploração da telefonia celular privada no interior de São Paulo .

par=ext917023-des-92a-2: Aproveitando o andamento rápido proporcionado pela lebre Jos Maes (2m48,57s aos 1000m, 5m40,36s aos 2000m) , Junqueira manteve-se sempre colado aos primeiros e aproveitou-se do facto para obter o mínimo olímpico que o ano passado falhara por tão pouco (então, 8m28,10s contra os 8m28,00 **pedidos**) .

<p>: Espero, Sr. Presidente, que os Vereadores do Município de Guarulhos usem de bom senso e votem dois **pedidos** de abertura de Comissão Especial de Inquérito -- CEI .

<p>: Merece referência, ainda, as chamadas condições de ação, as quais restam identificadas na possibilidade jurídica do **pedido**, na constatação do interesse processual e na verificação da legitimidade da parte interessada .

<p>: A Câmara de Vereadores de Pirajuí vota **pedido** de cassação do prefeito José Carlos Ortega (sem partido) na próxima quinta-feira .

Imagem 2. Linhas de concordância do lema *pedido* com anotação semântica de emoção:humildade

A partir das linhas de concordância, é possível observar que não há diferenciação nos sentidos independente da anotação, que foi inicialmente feita de forma automática. No entanto, é interessante percerber que há uma regra que exclui corretamente da anotação de emoção a expressão "a pedido de". Ainda assim, a leitura das linhas de concordância dos dois casos (com e sem anotação semântica) mostra que o sentido do lema *pedido* não é emocional em nenhum dos contextos. Ilustrativamente, a partir de uma busca[3] no corpus pelos complementos de *pedido*, as dez combinações mais frequentes foram "pedido de demissão"; "pedido de liminar"; "pedido de cassação"; "pedido de autorização"; "pedido de prisão"; "pedido de desculpa"; "pedido de impeachment"; "pedido de urgência"; "pedido de extradição"; e "pedido de falência". Dessas, a única combinação que indica emoção é "pedido de desculpa", que já faz parte do grupo "Humildade" por meio de uma regra de anotação do Emocionário.

4. Confirmação de Sentidos dos Lemas: pesquisas em dicionário ou ferramentas de busca na internet

O segundo momento da metodologia compreende os passos 4 e 5, que se constituem a partir de pesquisas em dicionários e/ou ferramentas de pesquisa on-line. Esses passos só precisam ser aplicados em casos nos quais não seja possível confirmar o uso do sentido emocional apenas pelas linhas de concordância, ou em casos nos quais o lema carrega muitos sentidos diferentes. O passo 4 consiste em procurar pela definição no dicionário para confirmar o uso com sentido de emoção equivalente ao do lema dentro dos contextos das linhas de concordância. Ao buscar pela definição do lema *pedido* no Dicionário Aulete Digital, também não pude confirmar o uso emocional do lema *pedido*.

[3] A expressão de busca correspondente é [lema="pedido" & sema=".*emo.*"] [lema="de"] @[pos="N"]

Ou seja, nem dentro dos contextos das linhas de concordância, nem isolado desses contextos que o corpus nos mostra, o lema carrega o sentido emocional. A vantagem de se pesquisar em dicionários é ter acesso a informação de usos daquela palavra em contextos que porventura não se apresentem na busca em corpus.

O passo 5 é a tentativa de encontrar usos semelhantes do lema sob análise e assim ratificar o uso daquele lema com sentido emocional pelo falante do português nas ferramentas de pesquisa on-line. Ele só acontece quando o dicionário não apresenta nenhuma definição emocional isolada, mas o contexto das linhas de concordância claramente indica emoção. Como no caso do lema *pedido* não houve menção de emoção, o passo 5 não teve utilidade. Vale ressaltar que o passo 5 só é aplicado na metodologia como um último recurso ao qual, na maior parte das vezes, não precisei recorrer. Quando utilizado na revisão de outros grupos, busco na ferramenta on-line pelas frases "X é sentimento" e/ou "X é emoção".

5. Resolução da Análise: manutenção, exclusão, adição ou transferência de lemas

O terceiro e último momento da metodologia une os passos 6 e 7. O passo 6 acontece ao longo da leitura das linhas de concordância nos passos 2 e 3. Nesse processo, observo coocorrência de outras palavras de emoção que possam ser inseridas neste ou em um dos outros grupos do Emocionário, caso apareçam em diversos exemplos. O grupo "Pena", por exemplo, ganhou os lemas *clemência, clemente, compassivo* e *condolência*. Após a observação das linhas de concordância na pesquisa lema a lema no processo de revisão do grupo, essas palavras coocorriam com os lemas em análise em muitos dos exemplos observados.

Já ao observar o lema *modéstia*, parte do grupo "Humildade", uma expressão se destacou do corpus ao longo da observação das linhas de concordância: "modéstia à parte". Essa combinação modifica o sentido de uso do lema. Então, será criada uma regra de anotação que indique que essa Expressão de Várias Palavras (EVP) com o lema *modéstia* sai do grupo "Humildade" para o grupo "Orgulho". Finalmente, no passo 7, mantenho ou retiro o lema previamente listado no grupo. No caso do grupo "Humildade", após a aplicação da metodologia foram retirados os lemas *pedido* e *pedir*, que serão discutidos na seção 6.

6. Discussão

Após a aplicação da metodologia, como explicitado nos passos 1 a 4, o lema *pedido* foi retirado do Emocionário por não carregar sentido emocional. O lema *pedir* também foi excluído pelo mesmo motivo. Além do sentido emocional não estar presente nos usos, ao estarem anotados como emoção, os lemas acabam por confundir resultados de buscas por anotação semântica de emoção em pesquisas que podem se interessar justamente por menção de emoção no corpus. Por exemplo, digamos que um pesquisador queira encontrar menção de emoção no texto de Machado de Assis. A Imagem 3 mostra que as menções do grupo "Humildade" aparecem entre as dez emoções mais recorrentes em sua obra.

Procura: **[autor="MacAss" & sema=".*emo.*"]**
Distribuição de **sema**
Corpo: OBras v. 10.18

56007 casos.

Distribuição

Houve **712** valores diferentes de **sema**.

emo:desejo	4371	emo:humildade	1486
emo:amor	4333	emo:medo	1135
emo:feliz	3294	emo:surpresa	990
emo:infeliz	2365	emo:feliz_emo:satisfeito	845
emo:amor_Hfam	2280	emo:vergonha	736
emo:gen	1734	emo:humildade_dizer	687
emo:amor_ac_H_am	1516		

Imagem 3. Distribuição semântica de emoção na obra de Machado de Assis

Procura: **[autor="MacAss" & sema=".*emo.*" & lema!="pedido|pedir"]**
Distribuição de **sema**
Corpo: OBras v. 10.21

54486 casos.

Distribuição

Houve **706** valores diferentes de **sema**.

emo:desejo	4371	emo:medo	1135
emo:amor	4333	emo:surpresa	990
emo:feliz	3294	emo:feliz_emo:satisfeito	845
emo:infeliz	2365	emo:vergonha	736
emo:amor_Hfam	2280	emo:feliz_am	658
emo:gen	1734	emo:alivio	653
emo:amor_ac_H_am	1516	emo:humildade	641

Imagem 4. Distribuição semântica de emoção na obra de Machado de Assis sem os lemas *pedido* e *pedir* anotados como emoção

Na Imagem 4 aparece o resultado das menções de emoção na obra de Machado de Assis excluindo-se os lemas *pedido* e *pedir*. A ocorrência de "Humildade" cai em quase setenta porcento, retirando o grupo das dez primeiras emoções, dando mais visibilidade a outros grupos de emoção, alterando a quantidade de menção de emoção na obra do autor e, por consequência, modificando bastante os resultados da pesquisa.

A carga semântica emocional de algumas palavras é mais forte que a de outras. O verbo *pedir*, por exemplo, passou a não integrar o Emocionário porque não faz necessariamente menção a uma emoção, ao passo que os verbos *suplicar* ou *implorar* significam "pedir com súplica, humildade" ou "pedir em uma situação de desespero". Ou seja, *suplicar* e *implorar* funcionam como a ação de *pedir* carregada com uma intenção emocional, como mostram os exemplos 01 e 02.

Exemplo 01: *id="A_Mortalha_de_Alzira Prosa:romance AA 1894 naturalismo_realismo_romantismo masc"*: Sua alma sangrava ainda, pedindo mais sacrifícios, e ele caía de joelhos, arranhando as carnes do peito com as unhas, e **suplicando** a Deus que lhe inspirasse um meio de resgatar-se, completamente, aos olhos da sua própria consciência envergonhada .

Exemplo 02: *par=4*: Mexeu de um lado para o outro com a cabeça, **implorando** com seus olhos para que tirassem aquele tubo, e a deixassem morrer em paz .

O verbo *pedir* também pode ter sua força semântica aumentada ou diminuída em contextos nos quais é combinado com outras palavras de emoção, mas não de forma isolada. Por isso é importante analisarmos a língua em contexto. A Imagem 5 mostra o que os falantes de português pedem no corpus OBras (Santos et al., 2018):

Procura: **[lema="pedir"] @[pos="N"]**
Distribuição de **lema**
Corpo: OBras v. 8.4

542 casos.

Distribuição

Houve **200** valores diferentes de **lema**.

licença	93	explicação	8	alvíssara	5	indenização	2	elogio	2
desculpa	28	informaçõe	7	justiça	4	ficha	2	nova	2
esmola	22	favor	7	compaixão	4	repetição	2	cerveja	2
notícia	20	coisa	7	amor	4	pouso	2	parati	2
água	15	silêncio	7	segredo	4	vênia	2	charuto	2
explicaçõe	14	café	6	meça	3	satisfaçõe	2	apoio	2
demissão	14	conta	6	permissão	3	juramento	2	esclarecimento	2
conselho	11	pão	6	proteção	3	alívio	2	providência	2
dinheiro	11	fogo	6	descanso	3	privilégio	2	agasalho	2
misericórdia	10	pousada	5	tempo	3	emprego	2	repouso	2

Imagem 5. Recorte dos complementos do lema *pedir*

A Imagem 5 faz um recorte dos cinquenta primeiros lemas de uma distribuição total de duzentos que complementam o verbo *pedir* no corpus OBras, composto por obras literárias brasileiras. Por esse recorte, já é possível perceber a carga emocional diferente entre os complementos de *pedir*: "pedir café" não faz menção à emoção já pela leitura distante do corpus. "Pedir misericórdia" faz menção à emoção pelo uso do lema *misericórdia*. Existe, portanto, um gradiente entre usos que exprimem claramente emoções e outros que não a exprimem.

7. Comentários Finais

A flexibilidade do projeto Emocionário se manifesta exatamente neste ponto de análise da semântica dos lemas do corpus. A observação e resolução não ocorrem de forma superficial, simplesmente por determinarem que cada lema esteja ou não no Emocionário, visto que a língua também não funciona dessa forma. Em alguns casos, podem ser feitas regras para a anotação específica no corpus. O lema *alegre*, que compõe os grupos "Felicidade" e "Satisfação", quando combinado com os lemas *Alto*, *Buriti*, *Córrego*, *Jardim*, *Monte*, *Porto*, *Pouso*, *Rio*, *River*, *Várzea* e *Vista* se refere a nomes de lugares. Logo, a anotação foi adaptada para que os casos em que *alegre* não se

refira a nomes de lugares não façam mais parte da anotação semântica de emoção em vez de ser excluído dos grupos do léxico.

Nos casos de verbos como *pedir*, uma possibilidade é analisar seus complementos tanto pelas linhas de concordância como pelas distribuições gramatical ou de lema, por exemplo, para identificar combinações recorrentes que façam menção de emoção e anotar especificamente esses casos, como no caso de "a pedido de", expressão na qual *pedido* não tem sentido emocional, e por isso não aparece anotado como parte do grupo "Humildade". O mesmo acontece no exemplo da expressão "modéstia à parte" na seção 5.

Tabela 1. Panorama quantitativo da revisão do léxico do Emocionário

Grupo	Lemas antes da revisão	Lemas excluídos	Lemas adicionados	Lemas adicionados a outros grupos	Lemas após a revisão
Desespero	40	4: arrebatar, arrebatar-se, desesperativo, insinuar.	8: angústia, angustiado, atormentado, atormentar desconfiado, desesperançoso, súplica, tormento.	11: amargura, atormentado, atormentar, tormento desconfiado, desconfiança, desconfiar mortificar súplica, suplicante, suplicar.	44
Esperança	36	1: crença.	2: crer, promissoramente.	2: entrever, augurar.	37
Humildade	24	2: pedido, pedir.	0	0	22
Pena	12	0	4: clemência, clemente, condolência, compassivo	0	16
Satisfação	37	1: agradado	6: apreciar, contentar, deleitar, deleitar-se, deleite, júbilo.	1: apreciar	42
Saudade	10	0	1: saudosismo.	0	11
Surpresa	41	1: súbito.	2: maravilhado, maravilhar.	0	42

Em termos quantitativos, a Tabela 1 elenca os números envolvidos na revisão, separados por "lemas antes da revisão"; "lemas excluídos"; "lemas adicionados"; "lemas adicionados a outros grupos"; e "lemas após a revisão". Como resultado final, os números não parecem tão distintos do cenário de antes da revisão. No entanto, por meio dessa revisão parcial do léxico foi possível mensurar a dificuldade de analisar aspectos semânticos dentro da língua com foco na descrição de emoção. A revisão também me permitiu delinear características e desafios da análise de emoção em português, aprimorar uma parcela da anotação do Emocionário e documentar uma metodologia que pode ser replicada futuramente nos grupos restantes ou em outras tarefas de revisão com objetivos semelhantes.

Referências

Mota, C. e Santos, D. (2015) Emotions in natural language: a broad-coverage perspective. In: Linguateca, https://www.linguateca.pt/acesso/EmotionsBC.pdf, Maio.

Santos, D, Freitas, C. e Bick, E. (2018) "OBras: a fully annotated and partially human-revised corpus of Brazilian literary works in the public domain", OpenCor, Canela, RGS, Brasil, Setembro.

Santos, D. e Bick, E. (2000) "Providing Internet access to Portuguese corpora: the AC/DC project", *Proceedings of the Second International Conference on Language Resources and Evaluation*, Editado por Maria Gavrilidou, George Carayannis, Stella Markantonatou, Stelios Piperidis e Gregory Stainhauer, LREC 2000, págs 205–210.

Santos, D., Simoes, A., e Mota, C. (2021) Broad coverage emotion annotation. In: Language Resources and Evaluation, No prelo.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Respostas emocionais da variação linguística: Análise exploratória de rastreio ocular

Raquel Meister Ko. Freitag[1], Julian Tejada[2], René Alain Santana de Almeida[1], Paloma Batista Cardoso[3], Victor Renê Andrade Souza[3], Vanesca Carvalho Leal[3]

[1]Departamento de Letras Vernáculas – Universidade Federal de Sergipe (UFS)
Cidade Universitária – 49.100-000 – São Cristóvão – SE – Brazil

[2]Departamento de Psicologia – Universidade Federal de Sergipe

[3]Programa de Pós-Graduação em Letras – Universidade Federal de Sergipe

{rkofreitag, jtejada, rene, palomabatista, victor.rene, vanescaleal}@academico.ufs.br

Abstract. *An exploratory study of exposure of participants to variants of a socially salient linguistic variable, progressive palatalization, using eye tracking is presented to examine the processing of linguistic variation. The results show that exposure to the stigmatized variant captured participants' attention and increased participants' pupil dilation, which can be interpreted as evidence of an emotional response.*

Resumo. *Um estudo exploratório de exposição de participantes às variantes de uma variável linguística saliente do ponto de vista social, a palatalização progressiva, foi realizado com o uso de rastreamento ocular, para examinar o processamento da variação linguística. Os resultados mostram que exposição à variante estigmatizada captou a atenção e aumentou a dilatação da pupila dos participantes, o que pode ser interpretado como evidência de uma resposta emocional.*

1. Introdução

Na observação de processos variáveis nas línguas, o papel do processamento tem ganhado relevo nos últimos anos, especialmente por conta da difusão de recursos tecnológicos não invasivos que permitem observar os efeitos de uma certa realização linguística não esperada para o contexto, em termos de respostas emocionais e demanda de atenção.

Uma das maneiras de observar se uma estrutura é mais saliente do que outra é quanto ao dispêndio de esforço de processamento. Na abordagem sociolinguística, no entanto, nem sempre formas salientes do ponto de vista cognitivo e de frequência são necessariamente salientes do ponto de vista social (e vice-versa), o que torna a mensuração do efeito de saliência muito difícil [Kerswill e Williams 2011, Kecskes 2011, Boswijk et al. 2020].

Diferentes abordagens têm sido utilizadas para observar efeitos de saliência no processo de variação linguística: estrutural, distribucional e sociocognitiva [Freitag 2018]. Os efeitos da estrutura linguística, como a saliência fônica e sua relação com a concordância, por exemplo, já são amplamente descritos no português. A saliência distribucional, implementada pelo monitor sociolinguístico [Labov et al. 2006,

Labov et al. 2011, Levon e Fox 2014, Freitag 2020], constructo para aferir aspectos perceptuais da variação linguística quantitativa em abordagens experimentais, como os estudos de percepção sociolinguística, tem revelado o quão sensíveis os falantes são às diferenças de frequência de uso de uma mesma variante.

O nível de consciência social desempenha um importante papel ao determinar quais variantes são sujeitas à correção [Labov 1972], e envolve o conhecimento implícito e explícito sobre a língua: perceber, reconhecer e processar a variação [Squires 2016]. Processos cognitivos ativados por um gatilho, como um traço sociolinguístico, ainda que controlados conscientemente, deixam pistas: a dilatação da pupila ou as expressões faciais [Freitag et al. 2020] também podem ser consideradas como evidências do julgamento social subjacente, que é a matriz do preconceito linguístico. Para a percepção sociolinguística, os efeitos da saliência sociocognitiva podem ser medidos na relação entre a saliência estrutural e a diferença na distribuição de variáveis que carregam indexação social e as que não carregam indexação social. Sendo um universal, os efeitos de superfície da saliência cognitiva apresentam um correlato fisiológico, um deles as respostas emocionais.

Emoções são respostas fisiológicas automáticas (como a dilatação da pupila [Partala e Surakka 2003]) ante determinados tipos de estímulos, e sensações são as experiências conscientes que acompanham essas mudanças fisiológicas. Estudos experimentais têm mostrado que as mudanças de excitação fisiológicas podem provocar experiências conscientes diferentes, a depender do contexto [Schachter e Singer 1962, Schachter e Wheeler 1962, Reisenzein 1983].

Diversas abordagens têm sido desenvolvidas para tentar se aproximar ao aspecto cognitivo das emoções usando percepção auditiva, visual, imaginação e/ou simulação de situações ou contextos [Lang e Bradley 2010, Löw et al. 2008, Bradley e Lang 1994, Bradley et al. 2006], com o desenvolvimento de diferentes técnicas para mensurar as respostas dos participantes, usando neuroimagem, mas também a observação do comportamento e a aplicação de instrumentos de avaliação. Tais procedimentos, quando evocam o paradigma do mundo visual, apresentam imagens ou textos com conteúdo que podem misturar a emoção com a experiência particular de um participante por apresentar um conteúdo linguístico com categorias semânticas fortemente arraigadas num contexto sociocultural; podemos, assim, transpor para o estudo de saliência e percepção sociolinguística que o estímulo auditivo ou visual a que um falante é exposto, a depender da variante linguística presente, pode evocar respostas emocionais.

O estudo do processamento da variação linguística tem se valido de medidas de rastreamento ocular, com resultados que evidenciam respostas emocionais após a exposição a um estímulo saliente [Foucart et al. 2019, Boswijk et al. 2020]. Neste texto, apresentamos um estudo exploratório em contextos de exposição a diferentes variantes de uma variável linguística saliente do ponto de vista social, a palatalização progressiva de oclusivas alveolares, a fim de subsidiar o desenvolvimento de técnicas *online* não invasivas, como o rastreamento ocular, para o estudo do processamento da variação linguística.

2. Palatalização

O processo de palatalização no português brasileiro resulta em realizações para /t/ e /d/ em adjacência a /i/: a realização oclusiva alveolar, e realizações palatais ou africadas.

Em contextos seguidos de /i/, como nas palavras “medida” e “batida” este processo é conhecido como palatalização regressiva. No entanto, quando o contexto é antecedido de /j/, como nas palavras “peito” e “doido”, também é possível ocorrer palatalização, em um processo denominado de palatalização progressiva.

A palatalização é um marcador dialetal do português brasileiro, com distribuição relativamente estratificada. Em comunidades em que o processo ainda é incipiente, como é o caso de Sergipe, estudos de produção evidenciam que o perfil de falante associado à variante palatal em contexto de palatalização regressiva é predominante de mulheres, mais jovens e mais escolarizados, residentes na capital. Tem se observado o aumento do uso da variante palatal, que é socialmente bem avaliada. Por outro lado, a variante palatal em ambiente de palatalização progressiva predomina na fala de homens, idosos e menos escolarizados, que residem no interior do estado. A variante palatal está em decréscimo, é socialmente mal avaliada, com comportamento de estereótipo negativo [Souza Neto 2008, Freitag 2015, Freitag e Souza 2016, Freitag e Santos 2016, Ribeiro e Corrêa 2018].

Em um estudo na comunidade considerando o constructo do monitor sociolinguístico [Freitag 2018], as realizações oclusiva e palatal dos dois ambientes de palatalização foram objeto de avaliação. O estudo considerou a exposição dos ouvintes-juízes a sequências de estímulos em áudio de supostas manchetes jornalísticas para um programa de rádio sobre saúde, gravadas por uma falante reconhecida como representativa da comunidade. O comando da tarefa experimental era julgar o grau de profissionalismo de cada um dos treinos de uma estudante de jornalismo candidata à locutora de rádio. Cada sequência de treino foi manipulada para conter diferentes gradientes de palatalização: 100% (todas palavras-alvo com realização palatal) - 70% - 30% - 50% - 30% - 0% (todas palavras-alvo com realização oclusiva).

Os ouvintes-juízes deveriam ouvir cada sequência de manchetes e avaliar o grau de profissionalismo do falante dos estímulos. Participaram do estudo falantes sergipanos (n = 304), estratificados quanto a sexo, idade, escolarização e zona de residência.

O resultado mostrou que o processo de palatalização regressiva não é sensível à frequência: com 100% ou 0% de realização palatal, a média da nota atribuída ao profissionalismo não apresenta diferença estatisticamente significativa. No entanto, a palatalização progressiva apresenta sensibilidade às frequências da variante [Freitag 2018].

Métodos experimentais (reação subjetiva, *matched guise*, julgamento de pares mínimos, etc) são usados desde os primeiros estudos da sociolinguística, sempre buscando minimizar os efeitos do Paradoxo do Observador. Em que pese o controle necessário para uma tarefa experimental, o estudo de processamento da variação linguística considera situações autênticas e minimamente invasivas. Nesse sentido, o rastreamento ocular configura-se como uma ferramenta metodológica com potencial para contribuir para a compreensão dos mecanismos cognitivos pelos quais a variação sociolinguística é processada, ao permitir explorar as ligações entre os diferentes níveis da gramática e informações sociais.

3. Método

3.1. Tarefa e participantes

O conjunto de dados desta análise foi utilizado em um estudo anterior [Freitag 2018], cujo objetivo era testar o paradigma do monitor sociolinguístico na variação da palatalização de oclusivas em uma variedade do Português Brasileiro. Paralelamente, foi coletada uma amostra em que ao mesmo tempo que os participantes (n = 18) eram expostos aos estímulos que consistiam em áudios com diferentes gradações de realização palatal, a tela do computador apresentava o texto escrito, com o objetivo de analisar os movimentos oculares nas zonas críticas, ou seja, as palavras-alvo da manipulação da palatalização. Para esta análise exploratória, consideramos apenas o processo de palatalização progressiva, cujo resultado no estudo de percepção de estímulos auditivos se mostrou sensível à diferença de frequências, sugerindo que o fenômeno é saliente.

3.2. Rastreio ocular

Para a coleta dos dados, foi utilizado um aparelho de rastreio ocular marca EyeTribe posicionado na parte inferior da tela do computador que controlava a apresentação dos estímulos, configurado para uma taxa de amostragem de 60Hz; a resolução da tela foi de 1024x768 *pixels* e a distância entre a tela e o participante foi de 57 cm em média. O software Opensesame [Mathôt et al. 2011] controlou tanto o procedimento de calibragem, que consistiu de 9 pontos, quanto os procedimentos de início e fim do registro do rastreador, assim como a apresentação dos estímulos visuais e auditivos do experimento do monitor sociolinguístico.

Os dados brutos de rastreio foram registrados e analisados usando versões adaptadas dos *scripts* pyGaze [Dalmaijer et al. 2014], disponíveis como um *fork* do repositório original no GitHub[1]. Os *scripts* operacionalizam os movimentos oculares em termos de fixações e sacadas que podem ser agrupadas por áreas de interesse (AdIs), ou áreas retangulares ao redor das quais se espera que se centralizem os movimentos oculares. As medidas obtidas em cada uma das AdIs são o número e a duração das fixações, número e duração das "entradas" nas AdIs, latência da primeira fixação e da primeira entrada. Adicionalmente foram construídos mapas de calor para observar globalmente a distribuição das fixações dentro e além das áreas de interesse.

As AdIs foram definidas como regiões retangulares em torno das palavras-alvo: *cuidado*, *muitos*, *oito*, *respeito* e *muito*. O tamanho de cada AdIs incluía, além da palavra em questão, o espaço entre linhas acima e embaixo da mesma, e o espaçamento antes e depois da palavra.

3.3. Pupilometria

O rastreador ocular registrou também o tamanho das pupilas dos participantes em séries temporais que foram também extraídas com as versões adaptadas dos scripts de pyGaze [Dalmaijer et al. 2014] e posteriormente processadas no *software* R [R Development Core Team 2009]. Essas séries representavam a oscilação no tamanho da pupila dos participantes quando observavam e escutavam cada um dos textos.

[1] `https://github.com/julian-tejada/PyGazeAnalyser`

Os dados brutos das séries temporais do tamanho da pupila foram processados aplicando dois filtros: o primeiro para controlar valores *outliers* na velocidade de dilatação, seguindo a fórmula apresentada por [Kret e Sjak-Shie 2019];

$$d'_{[i]} = max\left(\left|\frac{d_{[i]} - d_{[i-1]}}{t_{[i]} - t_{[i-1]}}\right|, \left|\frac{d_{[i+1]} - d_{[i]}}{t_{[i+1]} - t_{[i]}}\right|\right) \quad (1)$$

onde $d_{[i]}$ é o valor do diâmetro da pupila num determinado momento $t_{[i]}$, e a velocidade de dilatação $d'_{[i]}$ é calculada como o máximo valor absoluto normalizado entre a mudança relativa ao valor precedente $(i-1)$ e subsequente $(i+1)$. Já o segundo filtro eliminou as piscadas ou valores muito pequenos de tamanho de pupila (pupil size $<= 5$). Para todos os casos, os valores removidos foram substituídos por interpolação linear, e foram removidos os ensaios nos quais foi preciso interpolar mais do 20% dos dados.

Uma vez filtradas, as séries temporais foram normalizadas para controlar as diferenças individuais nos tamanhos da pupila. O processo de normalização min-max começou subtraindo de cada valor de cada série o menor valor encontrado na mesma, para posteriormente, dividir cada valor da série pela faixa de valores correspondente [Wendt et al. 2016].

Com as séries normalizadas, procedeu-se a centralizar os diferentes registros em torno do momento no qual cada grupo escutou uma das cinco palavras-alvo (*cuidado*, *muitos*, *oito*, *respeito* e *muito*), e usando o pacote ggplot2 [Wickham 2011] estimou-se uma linha de regressão para suavizar a série temporal usando modelos aditivos generalizados seguindo a proposta feita por [Boswijk et al. 2020].

3.4. Análise de dados

Os dados de rastreio ocular foram analisados usando modelos de ANOVA de 2 (realização sem palatalização, realização com palatalização) x 5 (*cuidado*, *muitos*, *oito*, *respeito* e *muito*) considerando ambos fatores como sendo de medidas repetidas. Quando necessário avaliar as diferenças entre os grupos foi utilizado o pós-teste de Bonferroni. Em todos os casos foram considerados estatisticamente significantes valores de $p < 0,05$, e tanto as análises quanto as figuras foram realizadas no R.

4. Resultados

4.1. Rastreio ocular

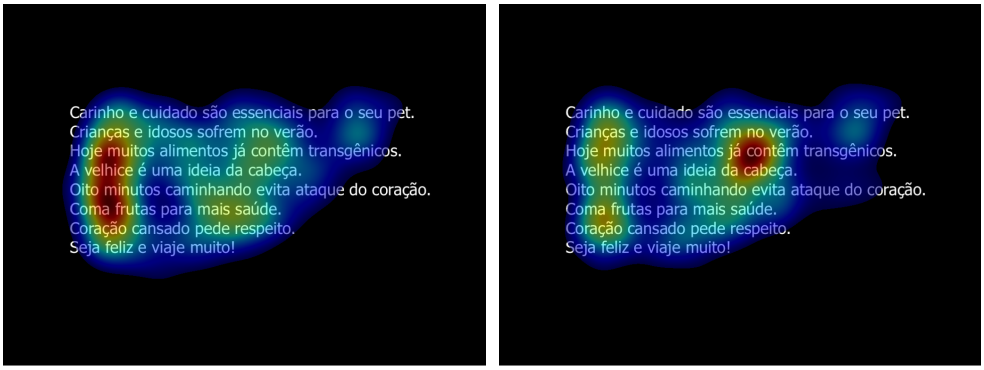
Os resultados da calibração mostram que a qualidade dos registros está dentro das margens de erro toleráveis [Ivanchenko et al. 2021], com uma exatidão de $xLeft = 0,4545 \pm 0,33$, $xRight = 0,4148 \pm 0,29$, $yLeft = 0,4545 \pm 0,33$, $yRight = 0,4148 \pm 0,29$ graus, e uma precisão de $X = 12,24 \pm 9,21$ e $Y = 12,24 \pm 9,21$ pixeis.

Os mapas de calor gerados a partir das frequências de fixações (Figura 1) consideram duas condições: a condição em que os estímulos auditivos das palavras-alvo tinham a realização com palatalização progressiva, que é a variante estigmatizada (a), e a que os estímulos auditivos das palavras-alvo tinham realização oclusiva, padrão, não estigmatizada (b). Dois padrões de fixação podem ser observados: na condição sem palatalização (b), a região que concentra as fixações no mapa de calor se aproxima da região de onde

anteriormente estava o ponto de fixação que aparecia na tela para iniciar o registro de cada ensaio. Já na condição com palatalização (a), as frequências das fixações concentram-se na margem esquerda do texto, na posição em que as palavras-alvo *muito* e *oito* aparecem.

Embora permita a visualização das frequências das fixações, o mapa de calor não permite a comparação dos efeitos entre as condições, o que pode ser realizado com a delimitação das AdIs (Figura 2), que permitem explorar a frequência de entradas (A) e a duração da fixação (B).

A exploração das AdIs mostra que a palavra *muitos*, nas duas medidas (A e B) e nas duas condições (com palatalização e sem palatalização) foi a que mais captou a atenção dos participantes; mesmo assim, a diferença das medidas nas duas condições é estatisticamente significativa. O mesmo ocorre com as demais palavras-alvo. A diferença entre as condições é maior na medida de duração das fixações do que na frequência das fixações.



(a) (b)

Figura 1. Mapas de calor gerados a partir das fixações nos estímulos (a) com palatalização e (b) sem palatalização.

4.2. Pupilometria

Para a pupilometria, selecionamos as duas palavras-alvo que apresentaram maior diferença nas médias nas duas condições na exploração das AdIs e visualmente presentes nos *heatmaps*: *muitos* e *oito*.

Os dados de mudanças no tamanho da pupila após os participantes terem escutado as palavras-alvo, em suas duas condições, apresentam a mesma dinâmica temporal reportada em estudos que avaliaram a dilatação da pupila como resultado de uma resposta emocional [Partala e Surakka 2003, Oliva e Anikin 2018], com variações que acontecem 400 ms após ter escutado o estímulo, o que permitem afirmar que houve uma resposta emocional. Na Figura 3 **A**, observa-se aumento acentuado no diâmetro normalizado da pupila na linha referente à condição com palatalização, evidenciando a dilatação. Em **B**, a dilatação é ainda mais acentuada.

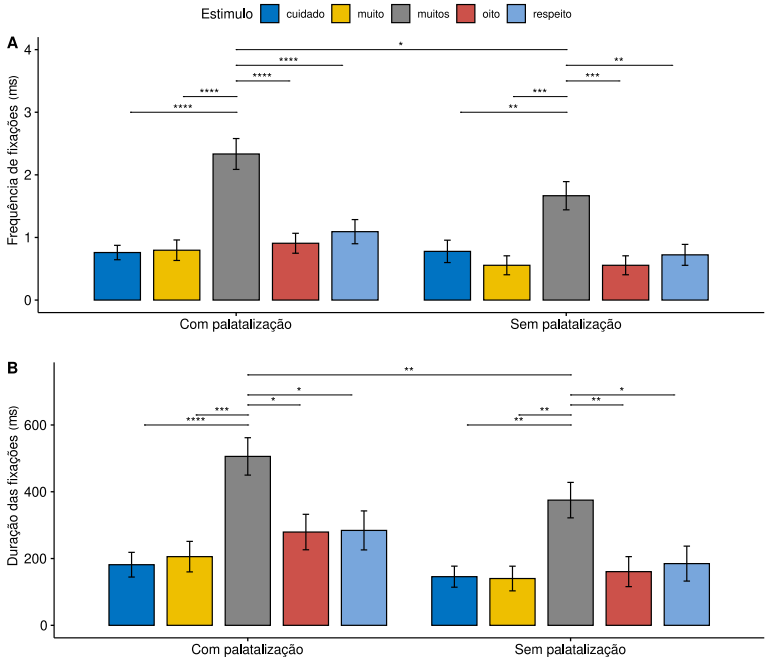


Figura 2. Frequência (A) e duração das fixações (B) em cada uma das AdIs para cada realização

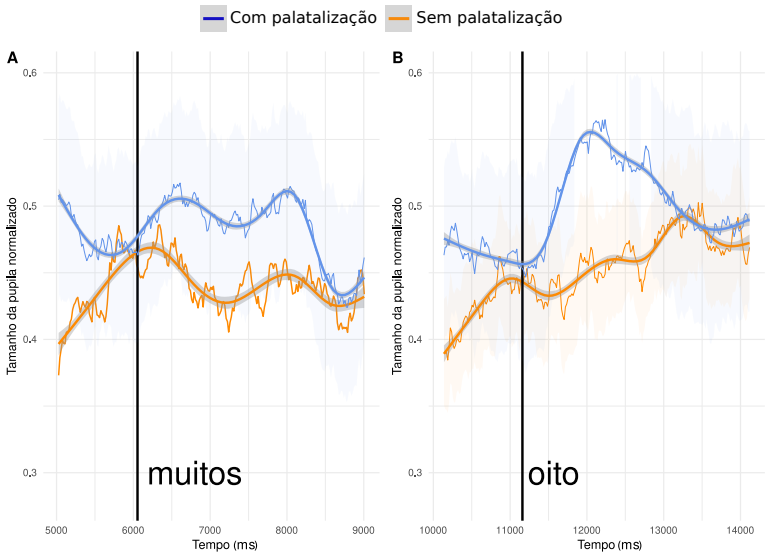


Figura 3. Tamanhos normalizados da pupila dos 18 participantes após terem escutado as palavras (A) muitos e (B) oito nas duas realizações

5. Discussão

O processo de palatalização progressiva é socialmente saliente e marcado, o que é evidenciado pela associação do uso a um perfil de falantes nos estudos de produção, e pelo julgamento negativo atribuído ao uso do traço com palatalização, nos estudos de percepção, configurando-se como uma variável do tipo estereótipo [Labov 1972]. Estereótipos são traços linguísticos que ativam comportamentos estigmatizadores, sendo alvo de preconceito. Assim, é de se esperar que haja uma resposta emocional ante a exposição a um estímulo com este traço.

A análise exploratória dos registros de rastreio ocular evidenciou que a exposição à realização com palatalização captou a atenção (como mostram os resultados do rastreamento ocular, com a medida de duração das fixações) e aumentou a dilatação da pupila (como mostram os resultados da pupilometria), o que podem ser interpretado como evidência de uma resposta emocional.

Estudos têm mostrado que estímulos emocionalmente carregados, sejam visuais ou auditivos, provocam aumento da dilatação da pupila, quando comparados com estímulos neutros [Partala e Surakka 2003, Nakakoga et al. 2020], e essa resposta estaria controlada pelo sistema nervoso autônomo simpático [Bradley et al. 2008], o que permite a utilização da mensuração da mudança na dilatação da pupila como uma medida da resposta emocional ante um determinado estímulo. O estudo de [Boswijk et al. 2020] considerou como alvo diferentes níveis gramaticais, em palavras isoladas; em nosso estudo, consideramos uma única variável, que estava inserida em um contexto linguístico maior (manchetes jornalísticas, cuidadosamente elaboradas para não conter outros traços variáveis salientes para a comunidade de fala onde a tarefa experimental foi realizada). Na direção apontada por [Boswijk et al. 2020], neste estudo exploratório observamos como uma forma linguística que é inesperada em um contexto (a realização palatal em ambientes de palatalização progressiva por uma universitária candidata à locutora de um programa de rádio), buscando evidenciar a relação entre o traço linguístico e o significado social atribuído pelos falantes. Durante a realização da coleta de dados, comentários informais dos participantes destacavam juízos de valor atribuídos à locutora dos áudios, do tipo "falando desse jeito não pode ser locutora de rádio".

A reação de dilatação da pupila pode ser consequência de diferentes gatilhos, inclusive das diferenças de iluminação, o que exige um ambiente de coleta de dados controlado. Nesse sentido, a coleta foi realizada em uma cabine de gravação, isolando potenciais oscilações de iluminação ou qualquer outra interferência. A diferença na proeminência da curva de aumento do tamanho da pupila entre as palavras-alvo *muitos* e *oito* pode ser explicada por um aspecto de frequência do processo de palatalização progressiva: em uma amostra com 10 horas de gravação de fala autêntica [Freitag 2015], foram identificados 775 contextos para palatalização progressiva, dos quais quase metade (n = 360) eram da palavra *muito* e suas flexões. Possivelmente, por conta da frequência, este item tenha um status diferenciado em termos de saliência.

Em relação ao rastreamento ocular, é necessário considerar que não houve planejamento para a disposição das palavras-alvo no texto; para a leitura alfabética, as posições mais à esquerda tendem a ser mais saliente do que a posição à direita; estes aspectos podem ter influenciado a incidência das fixações na palavra-alvo *muitos* à esquerda, mas não em *muito*, à direita e no final da linha.

Apesar das limitações, o estudo exploratório com a palatalização regressiva reforça a potencialidade de utilizar o rastreamento ocular como ferramenta para estabelecer parâmetros para identificação de variáveis socialmente salientes, contribuindo para resolver o problema da literatura com o conceito de "saliência", como mostram [Boswijk et al. 2020].

Referências

Boswijk, V., Loerts, H., e Hilton, N. H. (2020). Salience is in the eye of the beholder: increased pupil size reflects acoustically salient variables. *Ampersand*, 7:100061.

Bradley, M. M., Codispoti, M., e Lang, P. J. (2006). A multi-process account of startle modulation during affective perception. *Psychophysiology*, 43(5):486–497.

Bradley, M. M. e Lang, P. J. (1994). Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1):49–59.

Bradley, M. M., Miccoli, L., Escrig, M. A., e Lang, P. J. (2008). The pupil as a measure of emotional arousal and autonomic activation. *Psychophysiology*, 45(4):602–607.

Dalmaijer, E. S., Mathôt, S., e Van der Stigchel, S. (2014). PyGaze: an open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments. *Behavior Research Methods*, 46(4):913–921.

Foucart, A., Santamaría-García, H., e Hartsuiker, R. J. (2019). Short exposure to a foreign accent impacts subsequent cognitive processes. *Neuropsychologia*, 129:1–9.

Freitag, R., Tejada, J., Brito, Í. D. V., Pinheiro, B., Silva, L. S., Cardoso, P., e Souza, V. R. A. (2020). Estudo piloto da relação entre o julgamento de traços linguísticos e expressões faciais. *Cadernos de Linguística*, 1(2):01–19.

Freitag, R. M. K. (2015). Socio-stylistic aspects of linguistic variation: schooling and monitoring effects. *Acta Scientiarum. Language and Culture*, 37(2):127–136.

Freitag, R. M. K. (2018). Saliência estrutural, distribucional e sociocognitiva. *Acta scientiarum. Language and culture*, 40(2):e41173–e41173.

Freitag, R. M. K. (2020). Effects of the linguistics processing: Palatals in brazilian portuguese and the sociolinguistic monitor. *University of Pennsylvania Working Papers in Linguistics*, 25(2):4.

Freitag, R. M. K. e Santos, A. d. O. (2016). Percepção e atitudes linguísticas em relação às africadas pós-alveolares em sergipe. *A Fala Nordestina: entre a sociolinguística e a dialetologia. São Paulo: Blucher*, pages 109–122.

Freitag, R. M. K. e Souza, G. G. A. (2016). O caráter gradiente vs. discreto na palatalização de oclusivas em sergipe. *Tabuleiro de Letras*, 10(2):78–89.

Ivanchenko, D., Rifai, K., Hafed, Z. M., e Schaeffel, F. (2021). A low-cost, high-performance video-based binocular eye tracker for psychophysical research. *Journal of Eye Movement Research*, 14(3):10.16910/jemr.14.3.3.

Kecskes, I. (2011). Salience in language production. In *Salience and defaults in utterance processing*, pages 81–102. De Gruyter Mouton.

Kerswill, P. e Williams, A. (2011). "salience"as an explanatory factor in language change: evidence from dialect levelling in urban england. In Jones, M. C. and Esch, E., editors, *Language Change*, pages 81–110. De Gruyter Mounton.

Kret, M. E. e Sjak-Shie, E. E. (2019). Preprocessing pupil size data: Guidelines and code. *Behavior Research Methods*, 51(3):1336–1342.

Labov, W. (1972). *Language in the inner city: Studies in the Black English vernacular*. Number 3. University of Pennsylvania Press.

Labov, W., Ash, S., Baranowski, M., Nagy, N., Ravindranath, M., e Weldon, T. (2006). Listeners' sensitivity to the frequency of sociolinguistic variables. *University of Pennsylvania Working Papers in Linguistics*, 12(2):10.

Labov, W., Ash, S., Ravindranath, M., Weldon, T., Baranowski, M., e Nagy, N. (2011). Properties of the sociolinguistic monitor. *Journal of Sociolinguistics*, 15(4):431–463.

Lang, P. J. e Bradley, M. M. (2010). Emotion and the motivational brain. *Biological Psychology*, 84(3):437–450.

Levon, E. e Fox, S. (2014). Social salience and the sociolinguistic monitor: A case study of ing and th-fronting in britain. *Journal of English Linguistics*, 42(3):185–217.

Löw, A., Lang, P. J., Smith, J. C., e Bradley, M. M. (2008). Both predator and prey. *Psychological Science*, 19(9):865–873.

Mathôt, S., Schreij, D., e Theeuwes, J. (2011). OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, 44(2):314–324.

Nakakoga, S., Higashi, H., Muramatsu, J., Nakauchi, S., e Minami, T. (2020). Asymmetrical characteristics of emotional responses to pictures and sounds: Evidence from pupillometry. *PLOS ONE*, 15(4):e0230775.

Oliva, M. e Anikin, A. (2018). Pupil dilation reflects the time course of emotion recognition in human vocalizations. *Scientific Reports*, 8(1):4871.

Partala, T. e Surakka, V. (2003). Pupil size variation as an indication of affective processing. *International Journal of Human-Computer Studies*, 59:185–198.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Reisenzein, R. (1983). The schachter theory of emotion: Two decades later. *Psychological bulletin*, 94(2):239.

Ribeiro, C. C. d. S. e Corrêa, B. T. R. d. A. (2018). Avaliação social da palatalização de/t, d/em sergipe. *A Cor das Letras*, 19(4Especial):109–123.

Schachter, S. e Singer, J. (1962). Cognitive, social, and physiological determinants of emotional state. *Psychological review*, 69(5):379.

Schachter, S. e Wheeler, L. (1962). Epinephrine, chlorpromazine, and amusement. *The Journal of Abnormal and Social Psychology*, 65(2):121.

Souza Neto, A. F. d. (2008). Realizações dos fonemas/t/e/d/em aracaju sergipe.

Squires, L. (2016). Processing grammatical differences: Perceiving versus noticing. In Babel, A., editor, *Awareness and control in sociolinguistic research*, pages 80–103. Cambridge.

Wendt, D., Dau, T., e Hjortkjær, J. (2016). Impact of Background Noise and Sentence Complexity on Processing Demands during Sentence Comprehension. *Frontiers in Psychology*, 7:345.

Wickham, H. (2011). ggplot2. *WIREs Computational Statistics*, 3(2):180–185. _eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.147.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Complexidade textual em notícias satíricas: uma análise para o português do Brasil

Gabriela Wick-Pedro[1], **Roney L. S. Santos**[2]

[1]Universidade Federal de São Carlos (UFSCar), São Carlos, Brasil

[2]Universidade de São Paulo (USP), São Carlos, Brasil

gwpedro@estudante.ufscar.br, roneysantos@usp.br

Abstract. *This article presents an analysis of the textual complexity of satirical and true news for Brazilian Portuguese. The Fake has been a big problem nowadays. Satirical content is an important point in the automatic detection of false news as its use can cause underlying confusion in the analysis. To carry out this research, the NILC-Metrix tool was applied, and 16 measures were evaluated, including descriptive, syntactic and semantic aspects, noting a greater complexity for real texts.*

Resumo. *Neste artigo é apresentada uma análise da complexidade textual de notícias satíricas e verdadeiras para o português do Brasil. As chamadas Fake News – ou notícias falsas – têm sido um grande problema na atualidade. O conteúdo satírico é um ponto importante na detecção automática de notícias falsas, pois seu uso pode causar confusão subjacente na análise. Para realização desta pesquisa, foi aplicada a ferramenta NILC-Metrix e avaliadas 16 medidas, entre aspectos descritivos, sintáticos e semânticos, notando-se uma maior complexidade para os textos verdadeiros.*

1. Introdução

Historicamente, a capacidade de divulgação das notícias falsas está intrinsecamente ligada aos suportes de cada época, como os papiros e pergaminhos na Antiguidade, a criação da imprensa na década de 1430 e surgimento do rádio e da TV nos séculos XIX e XX, respectivamente. Na atualidade, com a Internet, a velocidade da comunicação cresceu e acelerou o tempo de disseminação das notícias, atingindo o maior número de pessoas em todo lugar do planeta em um curto espaço de tempo. As redes sociais como novos meios de comunicação permitiram um maior espaço para o indivíduo expor seus pensamentos, opiniões, emoções e posições de suas ideias, o que fortaleceu um comportamento individualista. Assim, o ambiente digital se tornou, com o passar dos anos, o habitat ideal para a disseminação da desinformação, pois é a partir das atuais tecnologias que uma notícia falsa é desenvolvida e compartilhada de uma forma sistêmica. O excesso de informação veiculada nas redes influenciou a sociedade atual, impactando diretamente em sua forma de pensar e agir no mundo contemporâneo.

Esse dinamismo das mídias sociais possibilita uma maior rapidez de leitura, o que resulta em baixo aprofundamento de grande parte dos textos veiculados na rede e desqualifica o leitor para uma compreensão de textos mais complexos. Logo, para realizar o processo de leitura, o leitor precisa compreender totalmente o texto e ter o conhecimento

prévio para obter integralmente o seu sentido [Leffa 1996]. Tal cenário colabora para minimizar a capacidade de entender argumentos mais difíceis da linguagem, de fazer uma análise crítica do que está sendo lido, além de favorecer na ascensão da disseminação de notícias falsas.

A heterogeneidade do termo e as diversas definições para o conceito de notícias falsas, ou popularmente, *Fake News*, transpassa as limitações conceituais, pois uma notícia pode ser projetada intencionalmente para enganar o leitor, ser criada para atrair cliques e obter lucro ou ser notícias satíricas com o objetivo de entreter [Rubin et al. 2015, Wardle and Derakhshan 2017, Tandoc Jr et al. 2018]. Grosso modo, Fake News pode ser definido como notícias imprecisas e, muitas vezes, fabricadas intencionalmente [Quandt et al. 2019].

Notam-se esforços acadêmicos recentes que procuram estudar conteúdo enganoso (do inglês, "*deception*"), averiguar o comportamento e o perfil dos usuários que compartilham e produzem esse tipo de notícia e como elas se espalham pela rede. Em particular, muitas frentes vêm sendo exploradas pelo Processamento de Língua Natural (PLN). Vale citar, por exemplo, o esforço para a língua portuguesa de [Monteiro et al. 2018] e [Silva et al. 2020], que fazem uso de características linguísticas para identificar automaticamente as *Fake News*.

Dentro desse contexto, as notícias satíricas podem criar dificuldades de entendimento e falsas crenças em leitores mais desatentos [Rubin et al. 2016]. Para a Literatura, a sátira é a representação literária de um estilo escrito em verso ou prosa que tem como objetivo a crítica às instituições, à sociedade e aos hábitos culturais de um povo. A sátira é considerada um gênero literário focado na crítica de um determinado tema, utilizando-se da ironia, do sarcasmo e da paródia para apontar falhas morais, políticas e sociais [Kreuz and Roberts 1993, Simpson 2003, Attardo 2014]. Para além da censura, a obra satírica acarreta entretenimento e leva o público ao riso por meio do absurdo, do exagero ou do ridículo.

Portanto, mostra-se relevante a tarefa de descrição e detecção automática de notícias satíricas. O presente trabalho propõe-se a investigar, com base em dados levantados por ferramentas de análise textual, especificamente o NILC-Metrix, como se dá a relação de inteligibilidade em notícias satíricas e verdadeiras para o português do Brasil, a fim de verificar as principais divergências linguísticas entre o conteúdo analisado, como aspectos lexicais, sintáticos e semânticos.

2. Métodos e Materiais

2.1. NILC-Metrix

O NILC-Metrix [Leal 2021] é um sistema computacional que contém por volta de 200 métricas propostas em estudos de discurso, psicolinguística, linguística cognitiva e computacional, que tem o objetivo de analisar a complexidade textual para o português.

O NILC-Metrix[1] está dividido em 14 categorias, que vão desde informações morfossintáticas e frequência de palavras até medidas mais robustas, como medidas psicolinguísticas e de legibilidade e facilidade de leitura do texto. A documentação do sistema,

[1]Disponível em http://fw.nilc.icmc.usp.br:23380/nilcmetrix

bem como as explicações de todas as métricas estão disponíveis online[2].

Segundo o autor, o NILC-Metrix pode ajudar os pesquisadores a investigar: (i) como as características do texto se correlacionam com a compreensão da leitura; (ii) quais são as características mais desafiadoras de um determinado texto, ou seja, quais características tornam um texto ou corpus mais complexo; (iii) quais textos têm as características mais adequadas para desenvolver as habilidades dos alunos-alvo; e (iv) quais partes de um texto são desproporcionalmente complexas e devem ser simplificadas para atender a um determinado público. Todos os pontos citados acima são úteis para o trabalho neste artigo, justificando a escolha do sistema para a análise dos textos satíricos e verdadeiros, os quais são explicados na seção a seguir.

2.2. O Corpus

O corpus desta pesquisa é composto por 300 notícias do domínio político, sendo 150 notícias satíricas e 150 notícias verdadeiras. As notícias satíricas foram extraídas automaticamente do site *Sensacionalista*[3], noticiário eletrônico que brinca com vários tópicos da política ou do entretenimento brasileiro. Já para as notícias verdadeiras, a coleta foi feita manualmente: primeiro palavras-chave foram identificadas e depois uma busca manual por cada notícia verdadeira equivalente à satírica. As características detalhadas podem ser observadas na Tabela 1.

NOTÍCIAS	QTD. NOTÍCIAS	TOKENS	TYPES	SENTENÇAS
Verdadeiras	150	107.133	11.304	5.721
Satíricas	150	22.963	4.843	1.212
TOTAL	**300**	**130.096**	**16.147**	**6.933**

Tabela 1. Características do corpus

Optou-se por não balancear o corpus para evitar a perda de informações na análise, uma vez que o número de palavras, sentenças ou diversidade lexical pode ser uma característica para a descrição desse tipo de conteúdo.

3. Resultados Obtidos

3.1. Índice Flesch

Considera-se o texto como um resultado parcial da comunicação do leitor com processos cognitivos, contextuais e linguísticos [Koch 1995]. Do inglês "*readability*", a leiturabilidade de um texto tem a finalidade de calcular o nível de facilidade de leitura do leitor. Nesse sentido, entende-se que o tamanho das sentenças e o vocabulário do leitor aumentam (ou diminuem) a capacidade de leitura de um texto [DuBay 2004].

Apesar de muitos estudos considerarem leiturabilidade e legibilidade como sinônimos, aqui, entende-se os dois conceitos de forma distinta. De acordo com [Resende and de Souza 2011], o termo leiturabilidade refere-se ao que está inserido no ato de ler, considerando o papel, a habilidade, as características, os conhecimentos e a experiência do leitor na atividade de leitura. Já legibilidade corresponde aos "elementos

[2]Disponível em http://fw.nilc.icmc.usp.br:23380/metrixdoc
[3]Disponível em https://blogs.oglobo.globo.com/sensacionalista

e recursos que o próprio texto, em sua materialidade, oferece ao leitor" [Resende e Souza 2011], i.e., relaciona-se com a facilidade se reconhecimento da forma das letras.

O Índice Flesch [Flesch 1979] procura uma correlação entre tamanho da sentença e o tamanho da palavra. A fórmula de Flesch, adaptada para o português por [Martins et al. 1996], é mostrada na Equação a seguir:

$$248,835 - 1.015\left(\frac{palavras}{sentencas}\right) - 84,6\left(\frac{silabas}{palavras}\right) \quad (1)$$

A Figura 1 apresenta a estatística da leiturabilidade das notícias satíricas e verdadeiras de acordo com o Índice Flesch. De acordo com a descrição da Tabela 2 e as informações apresentadas na terceira coluna do gráfico, as notícias verdadeiras são mais difíceis (87) em relação às satíricas (55). Ainda, em comparação, a primeira coluna do gráfico mostra que as notícias satíricas (4) são muito mais fáceis do que as notícias verdadeiras (1). A segunda coluna indica uma facilidade de leiturabilidade muito maior nos textos satíricos (90) em contraste com os textos verídicos (61).

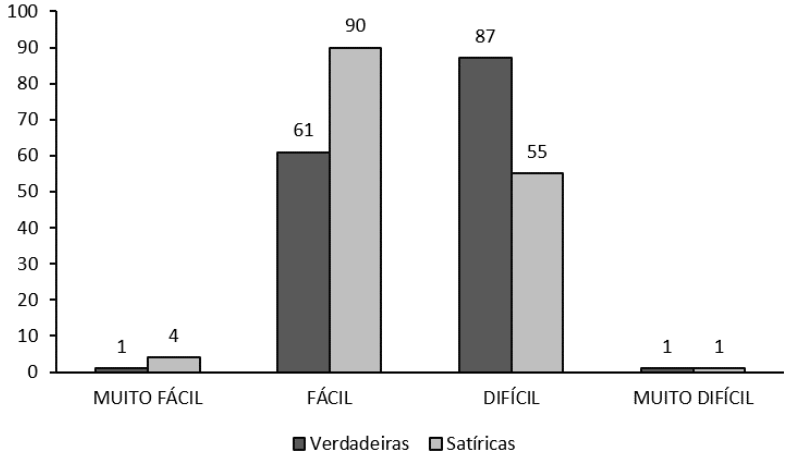


Figura 1. Estatística de leiturabilidade do corpus de acordo com o Índice Flesch Brasileiro

ESCORE	NÍVEL DE COMPLEXIDADE	GRAU ESCOLAR
100-75	Muito Fácil	1º a 5º ano
75-50	Fácil	6º a 9º ano
50-25	Difícil	Ensino Médio
25-00	Muito Difícil	Ensino Superior

Tabela 2. Escore do Índice Flesch Brasileiro: Nível de leiturabilidade por grau escolar

3.2. Avaliação da Complexidade Textual em Português do Brasil

Em geral, foram investigadas 200 medidas do NILC-Metrix, agrupadas em 14 categorias: *medidas descritivas, simplicidade textual, coesão referencial, coesão semântica, medidas psicolinguísticas, diversidade textual, conectivos, conectivos, léxico temporal, complexidade sintática, densidade de padrões sintáticos, informações morfossintáticas de palavras, informações semânticas de palavras, frequência de palavras e índices de leiturabilidade*. Entretanto, para a análise, foram selecionadas apenas as medidas com resultados mais relevantes, tais quais as medidas que tiveram valores distantes, bem como medidas importantes que são utilizadas em comparações de complexidade de texto, como o TTR [Templin 1957]. A Tabela 3 apresenta as médias de cada tipo de notícia quando aplicadas as medidas do NILC-Metrix.

CATEGORIA	*MÉTRICAS*	*VERDADEIRAS*	*SATÍRICAS*
MEDIDAS DESCRITIVAS	**Frequência de palavras de conteúdo**	611.825,41	490.304,19
	Número de palavras	594,08	156,44
	Número de sentenças	32,02	8,36
	Palavras por sentença	19,35	19,49
COESÃO REFERÊNCIAL	**Pronome anafórico do caso reto**	0,27	0,19
	Pronome demonstrativo anafórico	0,30	0,21
	Referência anafórica	1,55	1,05
	Referência anafórica (adjacente)	0,33	0,24
	Sobreposição de argumentos	0,70	0,97
	Sobreposição de argumentos (adjacentes)	0,84	1,06
	Sobreposição de radical de palavras	0,93	1,38
	Sobreposição de radical de palavras (adjacentes)	1,14	1,53
DIVERSIDADE LEXICAL	**TTR**	0,73	0,73
COMPLEXIDADE SINTÁTICA	**Incidência de orações subordinadas**	0,18	0,30
	Preposições por sentença	1,53	2,16
INFORMAÇÕES MORFOSSINTÁTICAS DE PALAVRAS	**Pronomes em 1ª Pessoa**	0,12	0,11
	Pronomes em 3ª Pessoa	0,71	0,49
	Verbos flexionados	0,25	0,38
	Verbos não flexionados	0,15	0,25
INFORMAÇÕES SEMÂNTICAS DE PALAVRAS	**Ambiguidade adjetival**	2,57	3,18
	Ambiguidade preposicional	0,28	0,49
	Ambiguidade verbal	9,99	10,22

Tabela 3. Características do corpus

Como esperado, os índices de medidas descritivas (*frequência de palavras de conteúdo, números de palavras, números de sentenças*) têm valores mais elevados em notícias verdadeiras, pois geralmente são textos mais longos e, consequentemente, mais complexos. Além disso, a ocorrência de pronomes em 3ª pessoa também é mais expressiva em notícias verdadeiras, o que pode indicar uma impessoalidade maior em textos reais em comparação às notícias satíricas.

Em relação aos mecanismos coesivos de referenciação textual, as medidas de referência anafórica – *pronome anafórico do caso reto, pronome demonstrativo anafórico, referência anafórica e referência anafórica (adjacente)* – mostram-se mais presentes em textos verdadeiros. Por se tratar de um recurso coesivo que busca a manutenção de sentidos apresentados anteriormente, quanto maior a métrica, maior a complexidade textual.

As medidas de *sobreposição de argumentos, sobreposição de argumentos (adjacentes), sobreposição de radical de palavras e sobreposição de radical de palavras (adjacentes)* possuem maior valor nas notícias satíricas. Entretanto, a repetição referencial

é uma característica de simplificação textual. Assim, quanto maior for o índice, menos complexo será o texto.

Por fim, nota-se que a ambiguidade lexical (*ambiguidade adjetival, ambiguidade preposicional e ambiguidade verbal*) é mais evidente em notícias satíricas. a ambiguidade é uma característica da construção de sentido da sátira e da paródia, uma vez que seu uso intencional causa no leitor uma confusão de sentido, podendo gerar um efeito de humor no texto. Contudo, para a compreensão do sentido ambíguo, é necessário o conhecimento extralinguístico daquilo que se lê e, dessa forma, quanto mais sentidos um texto tiver, maior será o esforço requerido do leitor para a desambiguação.

4. Conclusões

Em resumo, este artigo apresentou uma breve análise sobre a complexidade textual de notícias verdadeiras e satíricas para o português do Brasil por meio de métricas da ferramenta NILC-Metrix. Como já foi abordado neste artigo, sabe-se que compreensão de um conteúdo satírico está intrinsecamente ligada à dispositivos extralinguísticos e ao conhecimento de mundo do leitor. No entanto, aspectos linguísticos presentes na estrutura das notícias podem ser um indicativo da sátira na notícia, como a forte presença de ambiguidade lexical ser um elemento para a construção do humor.

Finalmente, como há poucas evidências de trabalhos que abordam a descrição de notícias satíricas, sobretudo para o português brasileiro, os índices aqui apresentados, mostram-se úteis não só para a descrição de notícias satíricas, como para a detecção automática de conteúdo enganoso. Além disso, os dados aqui discutidos podem ser usados futuramente como features linguísticas na implementação de classificadores automáticos de complexidade de textos jornalísticos utilizando algoritmos de aprendizado de máquina.

Agradecimentos

Os autores agradecem à CAPES (Código Financeiro 001), ao Escritório de Pesquisas da USP (PRP 668) e ao Centro de Inteligência Artificial (C4AI) da Universidade de São Paulo, apoiado pela IBM e FAPESP (nº 2019/07665-4)

Referências

Attardo, S. (2014). *Encyclopedia of humor studies*. Sage Publications.

DuBay, W. H. (2004). The principles of readability. *Online Submission*.

Flesch, R. (1979). How to write plain english: A book for consumers and lawyers.

Koch, I. (1995). O texto: construção de sentidos. *Organon*, 9(23).

Kreuz, R. J. and Roberts, R. M. (1993). On satire and parody: The importance of being ironic. *Metaphor and Symbol*, 8(2):97–109.

Leal, S. E. (2021). *Predição da complexidade sentencial do português brasileiro escrito, usando métricas linguísticas, psicolinguísticas e de rastreamento ocular*. PhD thesis, Universidade de São Paulo.

Leffa, V. J. (1996). *Aspectos da leitura*, volume 7. Sagra Porto Alegre.

Martins, T. B., Ghiraldelo, C. M., Nunes, M. d. G. V., and de Oliveira Junior, O. N. (1996). *Readability formulas applied to textbooks in brazilian portuguese*. ICMC-USP.

Monteiro, R. A., Santos, R. L. S., Pardo, T. A. S., de Almeida, T. A., Ruiz, E. E. S., and Vale, O. A. (2018). Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In *Computational Processing of the Portuguese Language*, pages 324–334.

Quandt, T., Frischlich, L., Boberg, S., and Schatto-Eckrodt, T. (2019). *Fake News*, pages 1–6. American Cancer Society.

Resende, N. R. and de Souza, A. C. (2011). A atividade tradutória e a relevância da leitura: legibilidade e leiturabilidade de textos humorísticos traduzidos. *Revista Gatilho*, 13.

Rubin, V. L., Chen, Y., and Conroy, N. K. (2015). Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Rubin, V. L., Conroy, N., Chen, Y., and Cornwell, S. (2016). Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second workshop on computational approaches to deception detection*, pages 7–17.

Silva, R. M., Santos, R. L., Almeida, T. A., and Pardo, T. A. (2020). Towards automatically filtering fake news in portuguese. *Expert Systems with Applications*, 146:113–199.

Simpson, P. (2003). *On the discourse of satire: Towards a stylistic model of satirical humour*, volume 2. John Benjamins Publishing.

Tandoc Jr, E. C., Lim, Z. W., and Ling, R. (2018). Defining "fake news" a typology of scholarly definitions. *Digital journalism*, 6(2):137–153.

Templin, M. C. (1957). Certain language skills in children; their development and interrelationships.

Wardle, C. and Derakhshan, H. (2017). Information disorder: Toward an interdisciplinary framework for research and policy making. *Council of Europe*, 27.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Constituintes Frasais com Função de Sujeito em Sentenças Judiciais

Ester Motta[1]; Maria José Bocorny Finatto[2]

Programa de Pós-Graduação em Letras – Universidade Federal do Rio Grande do Sul (UFRGS) – 91.501-970 – Porto Alegre – RS – Brasil

[1]estermottac@gmail.com; [2]mariafinatto@gmail.com

Resumo. *Descreve-se a organização sintática de um conjunto de Sentenças dos Juizados Especiais Cíveis, cujos documentos devem ser acessíveis ao cidadão leigo, sem auxílio de um advogado. São destacados os constituintes frasais com função de sujeito em 110 Sentenças à luz dos estudos de Terminologia, da Linguística de Corpus e de pesquisas sobre compreensão leitora. Verificou-se que a maioria desses constituintes assume formas que tendem a demandar maior sobrecarga na leitura e a exibir traços pouco coincidentes com padrões da linguagem cotidiana escrita, o que dificulta sua inteligibilidade para o público leigo.*

Abstract. *This work describes the syntactic organization of a set of Judgments of the Special Civil Courts, whose documents must be accessible to the lay citizen without the assistance of a lawyer. The phrasal constituents with subject function in 110 Judgments are highlighted in the light of Terminology studies, Corpus Linguistics and research on reading comprehension. The study found that most of these constituents assume forms that tend to demand greater burden in reading and to exhibit traits that hardly coincide with patterns of written daily language, which makes their intelligibility difficult for the lay public.*

Introdução

Este trabalho traz dados de um estudo-piloto realizado em uma pesquisa de doutorado em andamento junto ao Programa de Pós-Graduação da Universidade Federal do Rio Grande do Sul. A pesquisa procura gerar dados descritivos sobre a maior ou menor acessibilidade da linguagem em uso em Sentenças[1] dos Juizados Especiais Cíveis (JECs) do Poder Judiciário do Estado do Rio Grande do Sul (PJRS). O estudo-piloto concerne ao tema da complexidade sentencial e sintática desses documentos.

Os JECs foram criados em 1995, pela Lei n. 9.099, para fornecer uma resposta mais rápida da Justiça ao cidadão, com um trâmite processual menos complexo, dispensando-se a atuação de advogados em alguns casos. Os documentos produzidos por esses juizados, conhecidos como "tribunais de pequenas causas", deveriam ser acessíveis ao entendimento do cidadão leigo, prevendo-se que esteja desacompanhado de advogado.

Nesse contexto, os dados do estudo aqui apresentado têm por objetivo descrever

[1] Quando nos referimos às Sentenças dos JECs, escreveremos com iniciais maiúsculas, para diferenciá-las das outras acepções dadas a esta palavra, tal como o significado de *frase* e *sentença.*

a conformação sentencial e sintática de uma amostra de Sentenças dos JECS do PJRS, destacando-se a apresentação dos constituintes frasais com função de sujeito (doravante CFsSujeitos). A escolha pelo exame centrado nos CFsSujeitos deu-se porque, segundo Fulgêncio e Liberato (2010), autoras que tomamos como referência para o exame dos dados quanto à complexidade de leitura, a medida ou extensão desses constituintes da sentença, individualmente, é muito importante, pois formam unidades centrais para a construção do significado. Além disso, a sintaxe verificada nas frases dessas Sentenças nos parece, *prima facie*, bastante complexa, merecendo uma abordagem à parte. Para ilustrar essa impressão, trazemos o exemplo (1), a seguir. Trata-se de uma única frase com 101 palavras[2] e vários níveis de subordinação. O trecho em negrito representa um CFSujeito com 56 palavras.

> (1) De fato, embora não exista contrato expresso que demonstre o negócio jurídico realizado entre as partes, concluo que **os depósitos efetuados pelo réu em benefício do autor (fl. 34), aliados às fotos de fls. 42 e 43 e aos e-mails trocados entre as partes (fls. 28, 30 e 31), em especial, o e-mail de fl. 28 no qual o requerido confirma que os pagamentos seriam realizados de acordo com a realização da Obra, (sic)** corroboram a versão inicial, no sentido de que autor e requerido ajustaram a construção de uma casa, que seria paga de modo parcelado até o final da obra. (Processo: 9002991-14.2018.8.21.0029– RIO GRANDE DO SUL, 2019, grifos nossos.)

Assim, pretende-se estudar o *quanto* e *como* a organização sintática dessas Sentenças, a partir do constituinte escolhido (CFsSujeitos), tenderia a: a) influenciar a complexidade textual; b) contribuir para um *modus dicendi* peculiar dos discursos do domínio jurídico brasileiro. Objetivamos, assim, verificar, pontualmente, se os critérios norteadores das ações dos JECs[3] - oralidade, simplicidade, informalidade, celeridade - são atendidos pela configuração da linguagem empregada nos textos de suas Sentenças.

2 Referenciais teóricos e metodológicos

Nossa pesquisa envolve diversas áreas do conhecimento, como Linguística Textual, Psicolinguística, Linguística de Corpus, Estudos do Léxico, Terminologia, Acessibilidade Textual e Terminológica e Tradução Intralinguística. Valemo-nos também de ferramentas e métodos computacionais do Processamento da Linguagem Natural (PLN) para a coleta e descrição dos dados textuais.

Entre os estudos de PLN, servimo-nos dos importantes subsídios trazidos por Leal (2019), que analisou, extensivamente, a complexidade sentencial do português brasileiro escrito, usando métricas linguísticas, psicolinguísticas e de rastreamento ocular. Este pesquisador afirma que a complexidade de uma sentença está relacionada a fatores como: extensão da sentença, extensão dos seus elementos constituintes, ordem dos elementos da sentença, presença de anáforas, etc.

Ao analisar Sentenças dos JECs do PJRS, lidamos com a linguagem de um dos domínios do conhecimento humano, o Direito. Nos estudos linguísticos, os variados tipos de textos desse domínio já são analisados por diferentes teorias do texto e do discurso e

[2] Palavra, neste trabalho, é vista como uma unidade da língua escrita situada entre dois espaços em branco, ou entre espaço em branco e sinal de pontuação.
[3] Tais princípios encontram-se relacionados no art. 2º da Lei n. 9.099, de 1995.

também pela ótica da Terminologia. Entre os estudos terminológicos, filiamo-nos à Terminologia de perspectiva textual e comunicativa [Bourigault e Slodzian 2004; Krieger e Finatto 2004]. Isso significa que examinamos não apenas o léxico temático, mas toda a textualidade dos diferentes tipos de discursos especializados, o que inclui a semântica e a sintaxe textual e frasal, como o que analisamos nas Sentenças dos JECs.

2.1 Materiais e Métodos

Trabalhamos com 110 Sentenças do *corpus* de estudo (CE) do todo da pesquisa, que tem 440 Sentenças. São documentos exarados entre 2018 e 2019. A amostra, selecionada de modo aleatório, buscou incluir os diferentes temas neles tratados. A Tabela 1 caracteriza o CE e a amostra fixada.

Tabela 1. Dados do CE e da Amostra

	CORPUS DE ESTUDO (CE)	AMOSTRA
TOTAL DE TEXTOS	440	110
TOTAL DE FRASES	31.601,52	7.491
PALAVRAS POR FRASE	18,22	19,25
TOTAL DE *TOKENS*	543.407	137.700
TOTAL DE *TYPES*	15.388	8.477
HAPAX	5.258	3.239

A ferramenta utilizada para a geração do número de *tokens* (palavras do texto), de *types* (palavras diferentes do texto) e de *hapax legomenon* (palavras de frequência única no texto) foi o sistema AntConc 3.2.1w[4]. Esse é um *software* livre que fornece listas de palavras, *clusters*, listas de contextos, e outras utilidades para análise linguística. Para verificar o total de frases e o total de palavras por frase, utilizamos a ferramenta Nilc-Metrix[5] (Coh-Metrix 3.0, nova versão de 2020). As métricas oferecidas pelo NILC-Metrix analisam a inteligibilidade do texto e foram desenvolvidas em mais de uma década no NILC da USP, *campus* de São Carlos.

A Tabela 1 acima permite verificar que, em relação ao CE, a amostra apresenta percentuais próximos a 25% no que se refere a número de: textos (25%), *tokens* (24,3%) e frases (23,7%). O número de palavras por frase da amostra, por sua vez, é semelhante ao do CE, com apenas 1 palavra a mais. Em relação ao *hapax legomenon*, que evidencia as especificidades e as preferências lexicais do autor de um texto, o percentual de ocorrência ficou muito semelhante nos dois conjuntos de textos: 34,17% no CE (5.258 para 15.385 *types*); e 38,20% na amostra (3.239 para 8.477 *types*). Tais dados permitem concluir pela representatividade da amostra em relação ao CE nesses quesitos. Salientamos, porém, que esses dados não influíram na fixação prévia da amostra.

[4] Disponível em <http://www.laurenceanthony.net/software.html>
[5] Disponível em: <http://fw.nilc.icmc.usp.br:23380/nilcmetrix>

2.2 Análise e Processamento dos Dados

Nossa unidade de análise foi a sentença/frase demarcada por uma inicial maiúscula e ponto final. A partir de cada sentença, identificamos orações e seus constituintes. Entendemos como constituintes da sentença os elementos da oração que formam com o verbo uma estrutura de argumentos ou valência verbal.

No exame inicial da amostra, as análises sentenciais e oracionais foram manuais, sem um apoio computacional específico para a geração de árvores de dependência ou *parsing*. Isso foi feito com base no nosso conhecimento sobre análise sintática como linguistas com o propósito de gerar uma referência ou *goldstandard* para as análises automáticas com o todo do CE. Assim, lemos e classificamos cada uma das orações presentes em nossa amostra e delas extraímos os CFsSujeitos.

3 Resultados da análise e classificação dos CFsSujeitos

Após identificar os CFsSujeitos, classificamo-los da seguinte forma: a) pré-verbal (sujeitos colocados antes do verbo); pós-verbal (sujeitos colocados após o verbo); b) pronominal (sujeitos representados por pronomes – relativos, demonstrativos, retos, etc.); c) elíptico em duas situações – quando determinado pela flexão número-pessoa do verbo ou por sua presença em alguma oração antecedente -; e d) oracional (oração como sujeito de outro verbo).

A classificação quanto à posição do sujeito – pré ou pós-verbal – deu-se apenas em relação aos CFsSujeitos constituídos por sintagmas nominais.

Na Tabela 2, temos o total de CFsSujeitos selecionados com a quantidade encontrada para cada tipo.

Tabela 2. Ocorrências dos tipos de sujeitos da Amostra

TIPOS DE SUJEITOS					
	ELÍPTICOS	REALIZADOS			
		SINTAGMA NOMINAL		PRONOMINAL	ORACIONAL
		Pré-Verbal	3.869		
		Pós-Verbal	1.884		
			5.753	1.459	486
SUBTOTAL	4.159				7.698
TOTAL					11.857

Conforme a Tabela 2, os sujeitos realizados são os de maior ocorrência na amostra, porém os sujeitos elípticos representam 35,07% do total de CFsSujeitos da amostra. Além disso, os sujeitos pós-verbais estão em terceiro lugar no *ranking*. Tais dados permitem-nos fazer algumas inferências.

Estudos sobre compreensão leitora (Fulgêncio e Liberato 2010) já assinalaram que sujeitos pós-verbais e elípticos podem ser considerados fator de maior complexidade, pois requerem sobrecarga maior de memória de trabalho. Os pós-verbais, que não estão

na ordem canônica da língua (sujeito-verbo-complementos), vão contra um processamento natural da leitura. O leitor busca, em seus conhecimentos linguísticos, estruturas sintagmáticas que lhe sejam mais conhecidas e, em encontrando, este conhecimento prévio favorece a compreensão. É o que diz Coscarelli (2002, p. 14)

> Da mesma forma que existe um padrão silábico mais frequentemente encontrado em cada língua, existem também as estruturas sintáticas mais usadas. A estrutura mais simples do português é aquela em que se tem o sujeito seguido do verbo que, por sua vez, é acompanhado por um complemento.

E os sujeitos elípticos, por demandarem a busca pelo seu referente, tornam essa identificação mais complexa que a identificação de um sujeito explícito na oração. E isso sobrecarrega a memória de trabalho na leitura.

Após essa classificação inicial, adotamos dois procedimentos.

Procedimento 1:

Distribuímos os sujeitos pré-verbais, pós-verbais e oracionais por número de palavras para verificar a extensão das dependências sintáticas: o número de palavras entre a raiz sintática (no caso, o SV) e o dependente (no caso, o constituinte com função de sujeito). A Tabela 3 apresenta esses dados.

Tabela 3. CFsSujeitos por número de palavras

QUANTIDADE PALAVRAS	NÚMERO DE OCORRÊNCIAS		
	PRÉ-VERBAIS	PÓS-VERBAIS	ORACIONAIS
1 a 10	3654	1453	151
11 a 20	176	320	186
21 a 30	31	77	82
30 ou +	8	34	67
TOTAL	3869	1884	486

Pela Tabela 3, é possível verificar que a maioria dos sujeitos analisados fica na faixa de 1 a 10 palavras. A exceção está quanto ao sujeito oracional. E isso é previsível porque, em tese, a oração vai ter mais palavras que sintagmas nominais simples.

Apesar dessa alta incidência de sujeitos na faixa de 1 a 10 palavras, a maioria dos sujeitos pré e pós-verbais apresenta seus núcleos expandidos por encaixes oracionais, mais próximos do limite de 10 itens, como se vê nos exemplos a seguir.

(2) *Digressões abstratas [de que o procedimento esteja excluído]* não representa (sic) prova efetiva que autoriza a falta de cobertura securitária.

(3) Devido, pois, *o reembolso daquilo [que o contratante despendeu com o procedimento] cuja cobertura foi negada pela seguradora]].*

O exemplo (2) apresenta 8 palavras, sendo 6 representadas por uma oração relativa nele encaixada. Essa estrutura sobrecarrega a memória de trabalho do leitor, porque ele

precisa processar várias informações antes de chegar à principal. A falta de concordância verbal "não representa" ao invés de "não representam", evidencia essa situação. A memória de curto prazo, ao ter de processar a informação trazida pela encaixada *[de que o procedimento esteja excluído]*, não conseguiu manter o núcleo plural do sujeito – *digressões* -, e aparentemente relacionou o verbo com o sujeito da encaixada: procedimento.

No exemplo (3), o complemento do núcleo "daquilo" é uma oração relativa encaixada que, por sua vez, também apresenta uma palavra expandida por outra oração relativa encaixada, esta iniciada pelo pronome "cuja", pouco usual no cotidiano da Língua Portuguesa. Ou seja, é um constituinte complexo com vários encaixes hierarquicamente diferentes e apresenta também um nexo pouco frequente na língua. Segundo Fulgêncio e Liberato (2010, p. 141), "a quantidade de encaixamentos é outro ponto que causa a complexidade das estruturas e faz com que elas sejam mais difíceis de serem processadas".

O exemplo (4) reproduz um sujeito oracional.

(4) No caso concreto, verifica-se *que não se constata a verossimilhança das alegações da autora quanto à alegada falha por parte da empresa requerida quanto à perfectibilização de um plano em linha de telefonia móvel da autora diversa daquela [em relação à qual a consumidora pretendia fosse efetuado o plano], **assim como**, (sic) não se constata a verossimilhança da alegação da mesma no [que se refere ao fato [de que possuía duas linhas de telefonia móvel, tampouco junto à empresa requerida]]*.

O exemplo (4) traz um sujeito oracional composto relacionado pela conjunção *assim como.* A primeira oração, iniciada em *que não se constata*, apresenta 45 palavras. A segunda, iniciada com *assim como*, apresenta 31 palavras. O trecho todo é composto por 81 palavras organizadas em apenas 1 frase com vários constituintes de hierarquia, fator considerado dificultador para a compreensão leitora.

Procedimento 2:

Verificamos a forma como se apresentavam os sujeitos elípticos, os sujeitos oracionais e os sujeitos pronominais. A Tabelas 4 resume esses dados da amostra.

Tabela 4. Formas como se apresentam os CFsSujeitos elípticos, oracionais e pronominais na Amostra

SUJEITOS ELÍPTICOS		SUJEITOS ORACIONAIS		SUJEITOS PRONOMINAIS	
FORMAS VERBAIS	Oorrências	Como se apresenta	Ocorrências	Tipo de pronome	Ocorrências
Verbos na 3ª pessoa	1734	Oração desenvolvida (com conjunção)	265	Relativos (que, quem, qual)	1020
Verbos na 1ª pessoa	664	Oração reduzida	221	Demonstrativos	230
Verbos no Infinitivo	887			Retos	38
Verbos no Gerúndio	592			Mesmo[6]	103
Verbos no Particípio	282			Indefinidos	68
TOTAL	4159		486		1459

Quanto às formas dos sujeitos elípticos, os que se encontram com os verbos na 3ª pessoa são os de maior frequência em nossa amostra. Dentro desse grupo, 1.577 verbos estão na 3ª pessoa do singular e 157 estão na 3ª pessoa do plural. Estando o verbo na forma finita, em princípio a busca pelo seu referente fica mais acessível. Em relação aos verbos flexionados na 1ª pessoa, a grande maioria deles se refere ao redator do texto, sendo possível depreender essa informação pelo contexto situacional.

As outras formas verbais relativas a sujeitos elípticos são nominais. Abaixo apresentamos alguns exemplos.

(5) *Citada* (folha 83), a parte Requerida contestou o feito (folhas 81/104). *Alegou* que o poste da rede de energia foi instalado no local há anos [...]

No exemplo (5), a forma verbal "citada", num processo catafórico, refere-se ao sujeito "a parte requerida". Segundo Fulgêncio e Liberato (2010), a catáfora implica uma sobrecarga na memória de curto prazo, porque exige que o item reduzido tenha de ser guardado na memória até que seu referente apareça no texto. Isso, consequentemente, acarreta maior trabalho para o leitor e maior dificuldade na leitura.

E a forma verbal "alegou" retoma este mesmo sujeito por um processo anafórico. Ainda segundo as autoras Fulgêncio e Liberato (2010), a busca de um referente na anáfora pode prejudicar a legibilidade do texto. Se essa relação for transparente – os referentes sendo facilmente identificados -, a leitura segue sem problemas. Porém, havendo dificuldade de encontrar o referente, "a leitura pode ser atrasada ou até mesmo interrompida" [Fulgêncio e Liberato 2010, p. 85]. É o que ocorre no exemplo (6), a seguir.

[6] Colocamos "mesmo" separadamente, porque atua diversamente na amostra – ora como pronome reto, ora como pronome demonstrativo.

(6) Restou comprovado nos autos a culpa exclusiva do demandado, que invadiu a via preferencial pela qual a demandante trafegava, *desrespeitando* a sinalização e a preferencialidade da via [...]

O gerúndio "desrespeitando", destacado no exemplo (6), pode, numa leitura inicial, ser atribuído à "demandante", que consta na oração imediatamente anterior. Todavia, essa interpretação é incoerente com o contexto.

Como vemos na Tabela 5, a diferença entre as ocorrências relativas às orações reduzidas e desenvolvidas é de aproximadamente 9%. Cabe ressaltar que as orações reduzidas, por apresentarem o verbo no infinitivo em sua maioria sem flexão (apenas 2 situações de infinitivo flexionado) podem ser um fator dificultador para a leitura, visto que se torna mais difícil de encontrar um elemento textual ou situacional referente a ele.

Quanto aos sujeitos pronominais, pela Tabela 6, verificamos que os pronomes relativos (que, quem e qual) ocupam o primeiro lugar (69%) entre os sujeitos pronominais. Além disso, as 1020 ocorrências de pronomes relativos na função de sujeito indicam a existência de 1020 orações relativas encaixadas. Já mencionamos que as orações encaixadas agregam informação aos constituintes da oração anterior, o que demanda maior custo de processamento na leitura e, em princípio, oferece maior dificuldade para a compreensão.

A seguir, a Tabela 5 apresenta uma síntese dos CFsSujeitos encontrados na amostra.

Tabela 7. Síntese dos CFsSujeitos e suas principais características

TIPOS DE SUJEITOS POR ORDEM DE OCORRÊNCIA		PERCENTUAL EM RELAÇÃOAO TOTAL DE SUJEITOS	CARACTERÍSTICAS PRINCIPAIS
ELÍPTICO		35%	42% em formas nominais 37,91% em verbos na 3ª pessoa do plural
REALIZADO	Pré-verbal	33%	94% com até 10 palavras
	Pós-verbal	16%	77,12% com até 10 palavras
	Pronominal	12%	69% por pronome relativos
	Oracional	4%	45,48% orações reduzidas

Considerações Finais

Os dados encontrados nas análises deste estudo-piloto parecem comprovar que a linguagem empregada nas Sentenças dos JECs nem sempre atende aos princípios da art. 2º da Lei 9.099/95, entre eles os da simplicidade e da informalidade. Afinal, a simplicidade na linguagem, como vimos até aqui, não combina com períodos longos de 1 frase só com vários níveis de subordinação. Não combina também com um percentual de 63% de sujeitos (elípticos, pós-verbais e pronominais) que demandam maior sobrecarga na memória de trabalho do leitor.

Na continuidade de nossa pesquisa, confrontaremos os dados aqui apresentados com os de outros *corpora*. Buscaremos especialmente acervos de jornais populares (como o PorPopular: http://www.ufrgs.br/textecc/porlexbras/porpopular/). Assim, esperamos

verificar o quanto a apresentação desses CFsSujeitos aproxima-se ou não da linguagem cotidiana escrita em nosso país.

Referências

Bourigault, D. Slodzian, M. (2004) “Por uma terminologia textual”, In: Krieger e Araújo (orgs.). *Cadernos de Tradução*, Porto Alegre: Instituto de Letras, Universidade Federal do Rio Grande do Sul, p. 29-32.

Brasil. (1995) “Lei nº 9.099, de 26 de setembro de 1995”, Lei dos Juizados Especiais.

Coscarelli, C. V. (2002) “Entendendo a Leitura”, *Estudos da Linguagem*, Belo Horizonte, v. 10, n. 1, p. 7-27, jan./jun.

Fulgêncio, L. e Liberato, Y. (2010) “É possível facilitar a leitura: um guia prático para escrever claro”, São Paulo: Contexto.

Krieger, M. G. e Finatto, M. J. B. (2004) “Introdução à terminologia: teoria e prática”, São Paulo: Contexto.

Leal, S. E. (2019) “Predição da complexidade sentencial do português brasileiro escrito, usando métricas linguísticas, psicolinguísticas, e de rastreamento ocular”, 132 p., Monografia (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP.

Rio Grande do Sul. Poder Judiciário do Rio Grande do Sul. (2019) “Processo nº 9002991-14.2018.8.21.0029”, Comarca de Santo Ângelo, julgado em 23-09-2019.

Part III

Workshop de IC em Tecnologia da Informação e da Linguagem Humana

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Compilação de um corpus etiquetado da Língua Geral Amazônica

Dominick M. Alexandre, Juliana L. Gurgel, Leonel F. de A. Araripe

Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brasil

domimaia@alu.ufc.br, julianalgurgel@alu.ufc.br, leonel@daad-alumni.de

Resumo. *Este trabalho apresenta as etapas de compilação de um corpus da Língua Geral Amazônica (LGA), ou nheengatu, desenvolvido para a posterior implementação de um etiquetador morfossintático para o sintagma nominal dessa língua. O estudo representa um avanço na construção de banco de dados para línguas indígenas e na inclusão dessas línguas minoritárias no atual contexto científico e tecnológico. Os resultados confirmam a aplicabilidade do corpus compilado para etiquetadores e outros algoritmos de processamento de linguagem natural.*

Abstract. *This work presents the stages of compilation of a corpus of the Amazonian Lingua Franca (LGA), or nheengatu, developed for the implementation of a part-of-speech tagger for the noun phrase of this language. The study represents an advance in the construction of a database for indigenous languages and the inclusion of these minority languages in the current scientific and technological context. The results confirm the applicability of the compiled corpus to POS taggers and other natural language processing algorithms.*

1. Introdução

Combinando Linguística e Ciência da Computação, o processamento de linguagem natural (PLN) é tradicionalmente considerado uma subárea da Inteligência Artificial no Brasil. A princípio, essa tecnologia permite o tratamento computacional dos diversos níveis da linguagem humana e o aperfeiçoamento da comunicação entre humanos e computadores [Guinovart 2000]. Contudo, o trabalho descritivo empreendido pelo PLN pode, ainda, contribuir para a preservação de línguas em risco de extinção, como a Língua Geral Amazônica (LGA), ou nheengatu, cujos bancos de dados disponíveis e inserção no cenário tecnológico são ínfimos ou inexistentes. Nesse contexto, o presente trabalho apresenta as etapas da compilação de um *corpus* da LGA, visando sua implementação no desenvolvimento de um etiquetador morfossintático para o sintagma nominal desta língua [Alencar 2020].

Em seu período de máxima difusão, em meados do século XVII, a LGA era falada do Maranhão à fronteira brasileiro-peruana. Atualmente, o nheengatu é falado pelos habitantes da região do Alto Rio Negro, na Amazônia [Navarro 2012]. Segundo o banco de dados *Ethnologue*, existem aproximadamente 6.000 falantes de nheengatu no Brasil, 8.000 na Colômbia e um número ínfimo na Venezuela [Eberhard, Simons e Fennig 2021]. Em apenas cinco anos, o número total de falantes diminuiu de 19.060

falantes em 2016 [Lewis, Simons e Fennig 2016], para 14.000 em 2021 [Eberhard, Simons e Fennig 2021].

Uma vez que a existência de corpora anotados morfossintaticamente é condição obrigatória para o desenvolvimento de tecnologias de processamento automático de textos, esta pesquisa, que faz parte de um projeto de Iniciação Científica, ainda em andamento, representa um primeiro passo na construção de um banco de dados do nheengatu voltado para o PLN [Alencar 2020]. Com este trabalho, pretendemos contribuir para a visibilidade do nheengatu e fornecer, para a comunidade científica, um *corpus* útil ao desenvolvimento de ferramentas de PLN e de pesquisas em diferentes áreas das ciências humanas, especialmente a linguística e a literatura [Alencar 2021].

2. Fundamentação teórica

Este trabalho envolve duas áreas da linguística: a linguística computacional e a linguística descritiva. Uma vez que o objetivo principal deste trabalho é a compilação de um *corpus* do nheengatu para a etiquetagem morfossintática do sintagma nominal (SN) de sentenças em nheengatu, uma parte do arcabouço teórico norteador da pesquisa tem caráter descritivo e documental, enquanto outra parte tem caráter computacional.

Neste trabalho, consideramos as descrições da estrutura do SN e das classes de palavras conforme Cruz (2011), mas adotamos a terminologia das classes do nheengatu segundo Navarro (2011), devido ao seu aspecto formal e simplificador. À luz das duas descrições gramaticais, inventariamos as seguintes classes do SN do nheengatu: nomes, adjetivos, pronomes pessoais, indefinidos, quantificadores, demonstrativos e numerais. Combinados, estes trabalhos oferecem caminhos para lidar com desafios que emergem da classificação de palavras da língua, como a ocorrência da posposição *upé* (*em*, em português), que, em alguns casos, assume as formas variantes *-pe* e *-me* (esta sempre após vogal nasal), afixadas aos substantivos, por exemplo: (i) *kunhã-itá uiku paranã upé* (*as mulheres estão no rio*, em português); e (ii) *paranãme igara upitá ana* (*no rio a canoa ficou*, em português). Além disso, algumas palavras podem ter mais de uma etiqueta morfossintática, como *iuaté* (*alto*, em português), que pode ocorrer como substantivo ou adjetivo. Uma maneira de lidar com estes e outros desafios é implementar regras que modelem computacionalmente esses fenômenos.

A etapa computacional do trabalho diz respeito ao pré-processamento de textos eletrônicos e à construção de uma versão piloto do etiquetador. A primeira tarefa na construção de um *corpus* para etiquetagem consiste na tokenização, isto é, a segmentação do texto em unidades processáveis por máquinas, denominadas *tokens* [Mikheev 2004]. Um etiquetador morfossintático é uma ferramenta de PLN cuja função é atribuir uma etiqueta morfossintática a cada *token* (ou palavra) de um texto dado como entrada e retorna como saída o mesmo texto anotado morfossintaticamente [Jurafsky e Martin 2019]. Em geral, a construção dessa ferramenta pode ser feita a partir de duas abordagens: (i) baseada em dados ou probabilística, a mais comum, que consiste na utilização de corpora anotados para o treinamento de modelos com base no contexto e na frequência das etiquetas; e (ii) baseada em regras, que consiste na implementação de regras com base nas descrições gramaticais da língua [Voutilainen 2004]. Neste trabalho, utilizamos uma abordagem baseada em regras devido à inexistência de corpora anotados do nheengatu [Alencar 2020].

3. Metodologia

A compilação do *corpus* foi dividida em duas etapas: (i) a compilação dos textos e exemplos das lições do *Curso de Língua Geral* [Navarro 2011] a partir de um arquivo tokenizado do livro; e (ii) a compilação do glossário de Navarro (2011) em uma tabela passível de conversão para a estrutura de dados *Dictionary*, da linguagem de programação Python. Devido à complexidade da segunda etapa, esta foi subdividida em três partes: (i) revisão das classes de palavras; (ii) pré-processamento para extração; (iii) extração e finalização.

A compilação dos textos e exemplos de Navarro (2011) consistiu na extração manual das sentenças selecionadas e na sua compilação em arquivos de texto à parte, utilizando a codificação UTF-8. Em relação à compilação do glossário, na parte (i), revisamos as entradas lexicais do glossário e listamos verbetes cujas classificações eram ausentes ou incompatíveis com a descrição gramatical do livro ou com a versão mais recente do livro-texto de Navarro, publicada em 2016. Na parte (ii), realizamos a conversão de caracteres especiais, por meio de um programa, denominado "replace-char.py". Em seguida, modificamos ou adicionamos as classes de palavras nas entradas lexicais listadas na parte (i). Na parte (iii), extraímos as entradas lexicais que ocorrem no sintagma nominal do nheengatu. Em seguida, geramos tabelas de duas colunas por meio de um programa, denominado "tag-words.py". A primeira coluna contém a entrada lexical, que constitui uma chave e, a segunda, a etiqueta correspondente à sua classe gramatical, que constitui um valor atribuído à chave (ver Figura 1). Depois, expandimos a lista de nomes com as formas flexionadas no plural por meio de um programa, o qual denominamos "nominal-flexionizer-yrl.py". Assim, adicionamos o sufixo de plural a todos os substantivos definidos por Navarro (2011), isto é, neutros, masculinos e femininos.

Na Figura 1, temos um exemplo de como uma tabela de nomes se parece antes e depois do processamento pelo programa flexionador. Do lado esquerdo, apresentamos a tabela gerada pelo programa "tag-words.py" e, do lado direito, apresentamos a tabela gerada pelo "nominal-flexionizer-yrl.py".

Figura 1. Tabela de nomes do nheengatu antes e após o processamento pelo programa nominal-flexionizer-yrl.py.

Uma vez geradas, cada uma dessas tabelas pode ser convertida para uma estrutura de dados *Python Dictionary*. A construção da versão beta do etiquetador foi feita por meio da implementação de uma função capaz de aplicar essa estrutura de dados aos *tokens* de um texto recebido como entrada no etiquetador. Para cada *token*, é atribuída uma etiqueta, desde que a palavra esteja contida no dicionário; caso contrário, o programa retorna a palavra, sem anotação [Alencar 2020]. Para testar a ferramenta, realizamos um primeiro teste a fim de verificar a aplicabilidade do *corpus* compilado para utilização em tarefas de PLN e identificar os aspectos do algoritmo a serem

aperfeiçoados. No teste, a versão beta do etiquetador recebeu como entrada um arquivo contendo as sentenças do texto da primeira lição de Navarro (2011), que perfazem um total de 16 sentenças e 74 palavras, entre itens que ocorrem ou não no sintagma nominal.

4. Conclusões

Além do *corpus*, composto por sentenças do nheengatu extraídas de Navarro (2011), esta etapa da pesquisa resultou em mais dois produtos úteis à aplicação em tarefas de PLN, conforme Tabela 1: (i) um dicionário do tipo *Python Dictionary*, contendo as entradas lexicais do nheengatu e suas respectivas etiquetas morfossintáticas; e (ii) um conjunto de etiquetas das classes que ocorrem no sintagma nominal da LGA.

Tabela 1. Produtos da pesquisa

	DESCRIÇÃO	TOTAL
Tagset	Conjunto de etiquetas	18
Dicionário	Itens lexicais e suas *POS-tags*	522
Corpus	Sentenças em Nheengatu	726

Como resultado do teste, a versão beta do etiquetador obteve uma acurácia de 100%. Vale ressaltar, contudo, que todos os itens pertencentes ao sintagma nominal presentes no arquivo de teste constavam no dicionário utilizado e que a parcela do *corpus* testada representa um escopo bastante limitado do banco de dados. Portanto, a acurácia alcançada neste primeiro teste é meramente ilustrativa, servindo apenas ao propósito de verificar a aplicabilidade do *corpus* compilado para a construção de ferramentas voltadas para o PLN e de identificar os erros do algoritmo que precisam ser corrigidos a seguir.

Para trabalhos futuros, cumpre ampliar o banco de dados do nheengatu através da compilação de textos de outras obras, como Casasnovas (2006). Além disso, é preciso aperfeiçoar o algoritmo do etiquetador e testá-lo com relação ao restante do *corpus* compilado. Paralelamente, uma pesquisa de mestrado, ainda em andamento, objetiva a construção de uma ferramenta capaz de etiquetar sentenças inteiras do nheengatu [Gurgel 2021]. Todos os produtos da presente pesquisa, por sua vez, estão sendo gradualmente disponibilizados sob licença livre para comunidade acadêmica na plataforma GitHub.

Referências

Alencar, L. F. de. (2020). Projeto de pesquisa. “Técnicas em softwares livres para linguística de corpus (12ª Etapa)”. Fortaleza: Universidade Federal do Ceará. Não publicado.

Alencar, L. F. de. (2021). “Uma gramática computacional de um fragmento do nheengatu”. *Revista Estudos da Linguagem*, Belo Horizonte, v. 29, n. 3, p. 1717-1777.

Casasnovas, A. (2006). "Noções de língua geral ou nheengatú: gramática, lendas e vocabulário". 2. ed. Manaus: Editora da Universidade Federal do Amazonas; Faculdade Salesiana Dom Bosco.

Cruz, A. (2011). "Fonologia e Gramática do Nheengatú: A língua falada pelos povos Baré, Warekena e Baniwa". Utrecht: LOT.

Eberhard, D. M.; Simons, G. F.; Fennig, C. D. (org.). (2021). "Ethnologue: Languages of the World". 24. ed. Dallas: SIL International. Disponível em: http://www.ethnologue.com. Acesso em: 04 jul. 2021.

Guinovart, X. G. (2000). "Lingüística computacional". In: Ramallo, F.; Rei-Doval, G.; Yáñez, X. P. R. (org.). *Manual de Ciencias da Linguaxe*. Edicións Xerais de Galicia.

Gurgel, J. L. (2021). "Nheenga-Tagger: um etiquetador morfossintático para o nheengatu" (working title). Projeto de dissertação (Mestrado em Linguística) - Universidade Federal do Ceará, Fortaleza. Não publicado.

Jurafsky, D.; Martin, J. H. (2009). "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition". 2. ed. Upper Saddle River: Prentice Hall.

Lewis, M. P.; Simons, G. F.; Fennig, C. D. (org.). (2016). "Ethnologue: Languages of the World". 19. ed. Dallas: SIL International. Disponível em: http://www.ethnologue.com. Acesso em: 04 jul. 2021.

Mikheev, A. (2004). "Text segmentation". In: Mitkov, R. (Org.). *The Oxford handbook of computational linguistics*. Oxford, Oxford University Press, p. 209-221.

Navarro, E. D. A. (2011). "Curso de Língua Geral (Nheengatu ou Tupi moderno): A Língua das origens da civilização amazônica". São Bernardo do Campo: Paym Gráfica e Editora.

Navarro, E. D. A. (2012). "O último refúgio da língua geral no Brasil". *Estudos Avançados*, v. 26, p. 245-254.

Voutilainen, A. (2004). "Part-of-speech tagging". In: Mitkov, R. (Org.). *The Oxford handbook of computational linguistics*. Oxford, Oxford University Press, p. 219-232.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Ferramenta linguístico-computacional como facilitadora para o ensino de gramática na escola

Lívia Vicente Dutra[1], Natália Sathler Sigiliano[1]

[1]FrameNet Brasil - Faculdade de Letras

Universidade Federal de Juiz de Fora (UFJF)

Rua José Lourenço Kelmer, s/nº, Campus Universitário

36036-900 – Juiz de Fora – Minas Gerais – Brasil

livia.vdutralet@gmail.com, natalia.sigiliano@ufjf.br

Abstract. *The research "Genres, textual typologies and linguistic analysis: constitution of didactic resources for the contextualized work of linguistic knowledge in an approach guided by textual genres" aims to develop a linguistic-computational tool to assist professors in the approach to content of grammar based on textual genres, given that the relationship between textual genres and their linguistic-discursive construction still constitutes a challenge for the teaching of Portuguese language. This tool is built based on annotations from FrameNet Brasil. Through this resource, it is expected to make it possible for the user to search for textual genres that are more conducive to teaching certain grammatical topics, given their prominence in these genres.*

Resumo. *A pesquisa "Gêneros, tipologias textuais e análise linguística: constituição de recursos didáticos para o trabalho contextualizado dos conhecimentos linguísticos em uma abordagem orientada pelos gêneros textuais" objetiva desenvolver uma ferramenta linguístico-computacional de auxílio ao professor para a abordagem de conteúdos de gramática pautados em gêneros textuais, já que a relação entre gêneros textuais e sua construção linguístico-discursiva ainda se constitui em um desafio para o ensino de língua. Esta ferramenta é construída com base em anotações da FrameNet Brasil. Por meio dela, espera-se tornar possível ao usuário a busca por gêneros textuais mais propícios para o ensino de determinados tópicos gramaticais, tendo em vista sua proeminência nesses gêneros.*

1- Introdução

A pesquisa "Gêneros, tipologias textuais e análise linguística: constituição de recursos didáticos para o trabalho contextualizado dos conhecimentos linguísticos em uma abordagem orientada pelos gêneros textuais" foi motivada a partir de uma demanda frequente no Brasil quanto ao ensino de gramática na educação básica de forma contextualizada e pautado no uso de gêneros textuais. Tal necessidade se revela em práticas de ensino de gramática ainda arraigadas no nível da análise do período e da oração, especialmente no que diz respeito ao ensino de sintaxe e morfologia. Como forma de reaver tal relação, os professores de Língua Portuguesa (LP) buscam no nível da textualidade alternativas para a exploração dos efeitos de sentido no uso de determinadas categorias gramaticais disponíveis no texto. Entretanto,

concebe-se que "as escolhas linguístico-discursivas presentes num dado gênero não são aleatórias, mas ali estão para permitirem que um gênero funcione socialmente" (MENDONÇA, 2007). Dessa forma, a análise das estruturas micro textuais características de determinados gêneros textuais pode auxiliar nas escolhas de conteúdos gramaticais a serem desenvolvidos nas aulas de LP. Essa escolha, por vezes, tem ocorrido com base no repertório do professor ou do autor de material didático com relação aos seus conhecimentos anteriores com o gênero textual em estudo. Sendo assim, a pesquisa tem sido desenvolvida na FrameNet Brasil (FN-Br) com o objetivo de elaborar uma ferramenta linguístico-computacional que auxilie o professor da educação básica no ensino de LP, associando tipos e gêneros textuais a seus itens gramaticais mais proeminentes.

A prática de Análise Linguística (AL) prevê o relacionamento estreito com outras práticas de ensino aprendizagem de LP, ou seja, à leitura e à produção de textos (GERALDI, 1984; MENDONÇA, 2006; BRASIL, 1997; BRASIL, 2017). Contudo, diversas pesquisas revelam que os materiais didáticos e os professores de língua materna ainda encaram o ensino de AL como independente de outras práticas, travando uma relação de ensino de LP descontextualizado e afastado de práticas sociais. Ao mesmo tempo, muitos manuais de ensino e docentes já inovam suas práticas ao preverem aproximações bastante claras, como aquelas relativas ao estudo do modo imperativo a partir de textos instrucionais. Não há, no entanto, muitas destas associações reconhecidas e desenvolvidas no que tange aos elementos sintáticos e às categorias morfológicas (SIGILIANO e SILVA, 2019).

Esta pesquisa assume o propósito de auxiliar o professor da escola básica no cumprimento de abordagens mais recentes para o ensino de LP, a partir de um trabalho conjunto entre Linguística Computacional, Ciência da computação e Ensino de Língua Materna.

2- Metodologia

A pesquisa foi subdividida em quatro etapas, sendo elas: seleção de *corpus*, anotação de texto corrido, extração de padrões linguísticos dos textos e, por fim, elaboração da ferramenta.

Na primeira etapa, foram selecionados 25 textos de diferentes gêneros textuais, de caráter modelar, dentre os tipos argumentativo, descritivo, expositivo, injuntivo e narrativo. Buscou-se equilibrar a extensão dos textos. Estes textos compuseram o *corpus* de análise da pesquisa, que foi incorporado à base de dados da FN-Brl para que as anotações de texto corrido pudessem ser realizadas.

As anotações computacionais foram feitas na plataforma da FN-Br, denominada *Web Annotation Tool* (*WebTool*)[1], seguindo as diretrizes de anotação estabelecidas por Ruppenhofer et al (2016). O grupo de anotadores envolvidos foi composto por dez alunos da graduação que cursaram a oficina de anotação computacional, somando um total de 30 horas de dedicação no ano de 2019. Posteriormente, três bolsistas de Iniciação Científica (IC) envolvidos na pesquisa e treinados nas diretrizes de anotação fizeram as revisões e deram continuidade à tarefa de anotação de texto corrido, constante durante todas as etapas do trabalho. Uma anotação de texto corrido consiste na análise de cada uma das Unidades

[1] Link para acesso: http://webtool.framenetbr.ufjf.br/index.php/webtool/main

Lexicais (UL) constantes no texto, as quais evocam um determinado frame[2] cada. Essa análise é realizada em três níveis: o nível dos elementos de frame, o nível sintagmático e o nível das funções gramaticais, ou seja, ela é capaz de extrair e sinalizar dados sintáticos-semânticos. Um exemplo deste tipo de anotação pode ser observado na Figura 1, em que a UL *dedicar* foi anotada no frame Causar_fazer_progresso. Na primeira camada de anotação, foram marcados os elementos de frame *projeto* e *duração* de forma explícita e o *agente* a partir de uma Instanciação Nula Construcional (CNI), que indica que o sujeito é marcado pela desinência do verbo. Nas segunda e terceira, observam-se as funções e classificações gramaticais correspondentes. Esses dados são armazenados no sistema, possibilitando, por meio da elaboração de queries, a extração de padrões linguísticos, suas respectivas frequências e contextos de ocorrências relacionados aos gêneros textuais. Os itens gramaticais anotados foram nomes, verbos, adjetivos e advérbios.

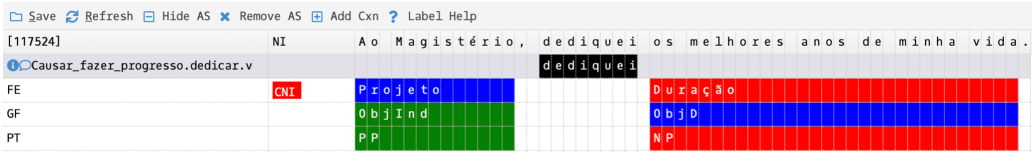


Figura 1 - Anotação da FrameNet Brasil em três camadas para a Unidade Lexical *dedicar* no frame Causar_fazer_progresso

A última fase envolve a elaboração da ferramenta linguístico-computacional, uma Interface de Consulta *online*. Ela ainda está em fase de teste e aprimoramento, por apresentar, por exemplo, dados com as nomenclaturas específicas à anotação na base de dados da FrameNet, mas já é possível fazer buscas e identificar alguns padrões linguísticos extraídos das anotações.

3- Resultados

A versão disponível da Interface de Consulta permite realizar buscas por *corpus*, tipos e gêneros textuais, abrangendo resultados semânticos e sintáticos, que possibilitam encontrar dados que evidenciam a relação dos gêneros com determinadas classes gramaticais ou padrões sintáticos (Marcuschi, 2002). A ocorrência, por exemplo, do modo imperativo em textos instrucionais aparece nos dados comprovando uma metodologia de ensino, em um primeiro momento, empírica. Ao escolher o gênero receita, observa-se que, por ter predominância de verbos no imperativo, os sujeitos são marcados como CNIs, ou seja, licenciados pela construção gramatical do imperativo. O mesmo não ocorre quando a busca é realizada pelo gênero notícia em que há predomínio de sujeitos oracionais, simples ou compostos, ou seja, expressos através de sintagmas nominais ou orações.

São estes modelos de resultados que se espera aplicar para todos os gêneros e tipos textuais disponíveis na Interface. No entanto, resultados gerados a partir da nomenclatura usada para as anotações não são funcionais para o professor da escola básica que, por vezes, desconhece tais termos por serem de uso muito específico da Semântica de Frames

[2] De acordo com Fillmore (1982) frames são "qualquer sistema de conceitos relacionados de tal forma que para entender qualquer um deles é preciso entender toda a estrutura em que ele se encaixa; quando uma das coisas em tal estrutura é introduzida em um texto ou em uma conversa, todas as outras estão automaticamente disponíveis."

(FILLMORE, 1982) e da Gramática de Construções (KAY & FILLMORE, 1999). Dessa forma, foi adicionada à tarefa de anotação uma quarta camada, denominada *sent* (sentença), que permite apresentar os padrões linguísticos encontrados em terminologias comuns ao contexto da escola, referentes à análise linguística, como é ilustrado na Figura 2, em que a UL *dedicar* é marcada como transitivo direto e indireto e seus complementos identificados. A inclusão da camada *sent* ocorreu na atual fase do projeto e, posteriormente, será adicionada à metodologia. A anotação da camada foi iniciada no ano de 2021 e está sendo realizada pelos bolsistas de IC da FN-Br. Estes dados serão disponibilizados na plataforma e, assim, o professor terá acesso a nomenclaturas que possam ser apresentadas a ele com mais clareza no que diz respeito à relação delas com os gêneros textuais com maior proeminência.

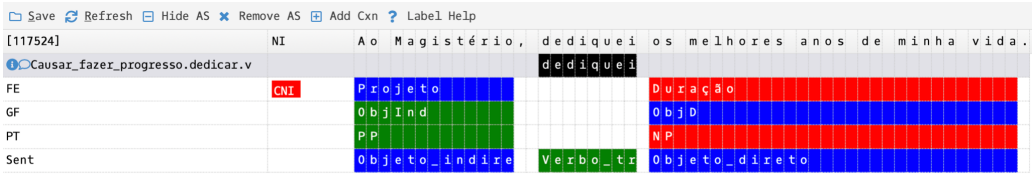


Figura 2 - Anotação da FrameNet Brasil para a Unidade Lexical *dedicar* no frame Causar_fazer_Progresso com a camada *sent*

Numa fase posterior, as anotações das quatro camadas realizadas sobre o *corpus* piloto serão usadas para treinamento de um algoritmo de aprendizagem de máquina de modo que, ao final do trabalho, todo e qualquer texto anotado no padrão FrameNet possa ser analisado automaticamente quanto aos elementos gramaticais tipicamente explorados no contexto escolar e, assim, possa passar a compor a Interface de Consulta.

Para além destes resultados, a pesquisa tem beneficiado não somente a elaboração de uma ferramenta *online* de busca de textos exemplares para que professores possam explorar competências de Análise Linguística de forma mais relevante para cada gênero textual, mas também enriquecido a base de dados da FN-Br. As anotações no *corpus* piloto geraram até o momento mais de 3500 UL anotadas em frames já existentes na base de dados da WebTool, além de muitas outras Unidades Lexicais que têm alavancado discussões sobre a criação de novos frames mais adequados para seus contextos.

4- Conclusões

Com base nos resultados preliminares expostos, já é possível comprovar a hipótese inicial da pesquisa que defende a associação de tipos e gêneros textuais a seus itens gramaticais mais predominantes, através de uma análise textual semântico-sintática e abstração de padrões linguísticos, aprimorando a metodologia de ensino de análise linguística. Dessa maneira, fica evidente que a Linguística Computacional pode ser uma grande aliada no ensino de língua, auxiliando professores em suas práticas pedagógicas.

Os próximos passos da pesquisa envolvem a finalização da anotação da camada *sent*, o aumento da base de dados, incluindo novos gêneros e mais exemplares de textos, o desenvolvimento do algoritmo de aprendizagem de máquina e o aprimoramento da interface de consulta para que esta se torne mais acessível e funcional para o seu usuário, o professor de Língua Portuguesa da educação básica.

Referências

Brasil.(2017). Base Nacional Comum Curricular. Brasília: MEC/Secretaria de Educação Básica. Disponível em: http://basbroenacionalcomum.mec.gov.br/. Acesso em: agosto de 2021.

Brasil.(1997). Parâmetros curriculares nacionais: introdução aos parâmetros curriculares nacionais. Brasília: MEC/SEF .

Fillmore, C.J. (1982). Frame semantics. In Linguistics in the Morning Calm, (ed. by The Linguistic Society of Korea), Soeul: Hanshin

Geraldi, J. W. (1984).O texto na sala de aula. Cascavel: Assoeste.

Marcuschi, L. A. (2002). Gêneros textuais: definição e funcionalidade. Gêneros textuais e ensino, 2, 19-36.

Mendonça, M.(2007) Análise linguística: refletindo sobre o que há de especial nos gêneros. In: Santos, C. F., Mendonça, M., Cavalcanti, M. C. B. (Orgs.). Diversidade textual: os gêneros na sala de aula (pp. 73-87). Belo Horizonte: Autêntica.

Mendonça, M.(2006). Análise linguística no ensino médio: um novo olhar, um outro objeto. In: Bunzen, C.; Mendonça, M. (Orgs.). Português no ensino médio e formação de professor(pp. 199-226). São Paulo: Parábola Editorial.

Ruppenhofer, J, Ellsworth, M. Petruck, M. Johnson, C. , Baker, C. And Scheffczyk, J. (2016). FrameNet II: Extended Theory and Practice. Disponível em https://framenet2.icsi.berkeley.edu/docs/r1.7/book.pdf. Acesso em outubro de 2020

Sigiliano, N. S.; Silva, W. R.(2017) Diagnóstico de propostas de análise linguística em livros didáticos aprovados em programa oficial. In: Magalhães, T. G., Reis, A. R. G.; Ferreira, H. M. (Orgs.). Concepção Discursiva da Linguagem, ensino e formação docente. (pp. 19-40). Campinas: Pontes.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Criação e Anotação do *corpus* de resumos científicos de Ciências Sociais Aplicadas

Sabrina de Fátima Barbosa Taniwaki, Jackson Wilke da Cruz Souza

Instituto de Ciências Sociais Aplicadas – Universidade Federal de Alfenas (UNIFAL-MG)
Varginha-MG – Brasil

{sabrina.tanikaki, jackson.souza}@unifal-mg.edu.br

Abstract. *With the growing interest and need in the creation of (semi)automatic tools that help the literacy process in academic textual genres, we present in this work the corpus of scientific abstracts in applied social sciences. Our objective was to study, at the sentential level, the rhetorical structure of the abstracts in the areas of Public Administration, Accounting and Economics, based on the typology proposed in the literature, using the WebAnno corpus annotation tool. As a result, (i) the organization of a set of 200 texts, (ii) the preliminary study of the rhetorical structure of those areas and (iii) the production of an annotation manual with specific guidelines on the identification of the rhetorical structure of the abstracts scientific.*

Resumo. *Com o crescente interesse e necessidade na criação de ferramentas (semi)automáticas que auxiliem o processo de letramento em gêneros textuais acadêmicos, apresentamos neste trabalho o corpus de resumos científicos em ciências sociais aplicadas. Nosso objetivo foi estudar, a nível sentencial, a estrutura retórica dos resumos das áreas de Administração Pública, Contabilidade e Economia, com base na tipologia proposta na literatura, por meio da ferramenta de anotação de corpus WebAnno. Tivemos como resultado, (i) a organização de um conjunto de 200 textos, (ii) o estudo preliminar da estrutura retórica das referidas áreas e (iii) a produção de um manual de anotação com diretrizes específicas sobre a identificação da estrutura retórica dos resumos científicos.*

1. Introdução

Os Gêneros Textuais (doravante, GTs) são "textos materializados que encontramos em nossa vida diária e que apresentam características sociocomunicativas definidas por conteúdos, propriedades funcionais, estilo e composição característica" [Marcuschi 2002, p.4]. Segundo Marcuschi [2002], só conseguimos nos comunicar por GTs, que surgem emparelhados às nossas necessidades comunicativas. Nesse sentido, o ensino-aprendizagem dos gêneros estaria condicionado à experiência linguística com que o falante tem com cada um deles.

Essas concepções norteiam, há alguns anos, o ensino de GTs tanto no ensino básico, bem como no ensino superior. Por estarem conectados a situações comunicativas, os gêneros que ficam à disposição dos alunos variam de acordo com o nível de letramento de cada um deles. Vieira e Faraco [2019] classificam os gêneros em função da formalidade (informal, semiformal, formal e ultraformal) inerente ao contexto sociocomunicativo em que emergem, como os gêneros acadêmicos.

Partindo desse princípio, as dificuldades enfrentadas pelos alunos no processo de letramento de GTs, especificamente o acadêmico, são caracterizadas por conta de os alunos não terem participado anteriormente de situações comunicativas que exigissem

determinados gêneros. Isso não quer dizer que o aluno-aprendiz não domine a língua e suas regras de funcionalidade [Bakhtin e Volochinov 2006].

Para mitigar as dificuldades de letramento de GTs acadêmicos, os ambientes computacionais de auxílio à escrita disponibilizam ao usuário modelos de estruturas retóricas de GTs, assistindo-o na organização e produção textual, como o SciPo [Antiqueira *et al.* 2003]. Apesar de abordarem gêneros estabilizados e que sofrem pouca variação (como os resumos acadêmicos), o fato de esses ambientes terem sido construídos dependentes de domínio (Computação, Física e Farmácia, por exemplo), faz com que a aprendizagem do gênero possa ser insuficiente.

Visando contribuir em ampliar a disposição de subsídios linguísticos às áreas de Letramento acadêmico e de Processamento de Línguas Naturais (doravante, PLN), apresentamos neste trabalho a construção de um *corpus* de resumos científicos das áreas de Administração Pública, Contabilidade e Economia. Para tanto, os textos foram anotados semi automaticamente a nível sentencial de acordo com a tipologia da estrutura retórica proposta por Feltrim [2004] e Feltrim *et al.* [2004].

Este artigo está organizado em cinco seções, além desta Introdução. Na Seção 2, apresentamos trabalhos que se relacionam a esta proposta de pesquisa, os quais investigaram estruturas retóricas de GTs. Na Seção 3 apresentamos a metodologia deste trabalho, bem como a caracterização do *corpus* organizado. Na Seção 4 demonstramos os resultados e a discussão, além de considerações finais, na Seção 5.

2. Trabalhos relacionados

Os trabalhos mais recentes que investigaram a estrutura retórica (como Iriguti e Feltrim [2019] e Teufel e Moens [2002]), baseiam-se em métodos de aprendizado supervisionado, portanto, dependentes de conjuntos de textos dos quais se possam extrair informações e conhecimentos linguísticos.

Para o Português do Brasil (PB), destacamos o trabalho de Feltrim *et al.* [2004], em que foi proposto o *Argumentative Zoning for Portuguese* (AZPort). Após o estudo no CorpusDT, composto por resumos científicos da área de Ciências da Computação, os autores apresentaram o conjunto de seis macrocategorias que caracterizam a estrutura retórica de resumos científicos nessa área. Tais macrocategorias são *Contexto* (55,7% de frequência), *Lacuna* (42,3%), *Propósito* (100%), *Metodologia* (63,4%), *Resultado* (67,3%) e *Conclusão* (30,7%); cada uma dessas categorias subdividem-se em mais três microcategorias.

Feltrim *et al.* [2004] anotaram manualmente todas as sentenças no conjunto de 52 textos (que somam 366 sentenças), e concluíram que nem todas as categorias são necessárias para compor um resumo científico. Ademais, apesar de o modelo retórico prever uma ordem com que essas categorias ocorrem, ela deve ser compreendida como diretriz e não como regra; isso quer dizer, que a sequência de ocorrência não é fixa, corroborando o posicionamento de Vieira e Faraco [2019] quanto à plasticidade dos GTs.

Tomando o trabalho de Feltrim *et al.* [2004] como base para a descrição da organização retórica de resumos em PB, é necessário ressaltar a importância da tarefa específica de anotação de *corpus*. Pustejovsky e Stubbs [2012] apontam que um dos papeis do PLN é capturar propriedades das estruturas linguísticas que possam ser

aprendidas por sistemas computacionais. Para que os algoritmos aprendam de maneira eficiente e eficaz, a anotação e identificação de tais propriedades devem ser precisa e relevante.

3. Metodologia

Metodologicamente, este trabalho foi realizado segundo a síntese das etapas de anotação de *corpus* propostas por Hovy e Lavid [2010], a saber: (i) Elaboração do *corpus*, (ii) Criação do manual de anotação e (iii) Anotação e avaliação dos *subcorpora* (estudo e treinamento). Tais tarefas são apresentadas respectivamente nas subseções, a seguir.

3.1. Elaboração do *corpus*

Partindo do pressuposto teórico de Sardinha [2004], decidimos que o *corpus* deveria ter o seguinte *design*: *Modo-Escrito* (composto por textos escritos); *Tempo-Diacrônico* (composto por resumos publicados entre 2009 e 2019); *Seleção-Equilibrado* (composto por resumos de periódicos de Qualis CAPES elevada quando possível); *Conteúdo-Especializado* (resumos científicos/acadêmicos); *Autoria-De língua nativa* (autores falantes de português); Finalidade (*subcorpora* de estudo e de treinamento); *Tamanho-Médio-grande* (200 textos, 1.118 sentenças e 35.904 palavras); *Anotação-Corpus anotado* (nível sentencial).

As áreas do conhecimento que compõem o *corpus* são Contabilidade/Economia (73 textos), Administração Pública (97 textos) e Economia (30 textos). A coleta e armazenagem dos textos durou cerca de um mês.

3.2. Criação do manual de anotação

A partir da elaboração do *corpus*, desenvolvemos o manual de anotação[1], o qual contém as diretrizes que nortearam este trabalho. Para tanto, estudamos em 1/4 do *corpus* (a saber, 50 resumos) o *tagset* proposto por Feltrim *et al.* [2004]. Esse estudo inicial foi feito por três anotadores, após quatro meses de análise da proposta dos autores, resultando numa anotação preliminar.

Utilizamos as macro e microcategorias propostas pelos autores, pois, ao longo dos estudos iniciais, percebemos que poderíamos identificar características relativas às próprias (sub)áreas de Ciências Sociais Aplicadas, e essas em relação às Ciências da Computação. Assim, na versão final do manual apresentamos as etiquetas utilizadas em nosso trabalho, a definição de cada uma delas, exemplos de sentenças anotadas, e instruções técnicas para utilização do ambiente virtual de anotação.

3.2. Anotação e avaliação dos *subcorpora*

Nesta tarefa, anotamos o *subcorpus* de treinamento, que corresponde a 3/4 do conjunto completo, totalizando 150 resumos. A anotação durou quatro meses, com uma equipe de dois anotadores. Para tanto, utilizamos o ambiente virtual WebAnno [CASTILHO *et al.* 2016]. Trata-se de um ambiente que reúne um conjunto de ferramentas relativas à tarefa de anotação de *corpus*, como a possibilidade de escolha da granularidade da anotação (palavra, sintagma, sentença etc.) e a disponibilidade de ferramentas de estatística textual (para medir concordância, tokens e types etc.), por exemplo.

[1] Disponível em: https://github.com/jackcruzsouza/EstruturaRetorica

Além disso, outra vantagem oferecida pelo referido ambiente de anotação é o fato de ser colaborativo, fazendo com que haja diferentes papeis entre a equipe do projeto (como moderador, anotador, gerente etc.), sem que haja necessidade de instalar alguma ferramenta e/ou ter conhecimento prévio de programação. O formato de saída do arquivo é em XML, e é compatível com a ferramenta Brat [Stenetorp *et al.* 2012].

Quanto à avaliação, obtivemos 60% e 72% de concordância nos *subcorpora* de estudo e de treinamento, respectivamente, segundo a medida Kappa, calculada automaticamente no próprio ambiente WebAnno.

4. Resultados e discussão

A partir da observação da anotação do *corpus*, foi possível tecer algumas considerações acerca das categorias retóricas.

Com relação ao *Contexto*, os resumos introduzem mais brevemente o assunto abordado no trabalho, além de explicação de alguns termos técnicos para melhor entendimento do leitor. Sobre a categoria *Propósito*, os resumos da área de Ciências Sociais Aplicadas apresentam, na maioria das vezes, apenas um objetivo principal do trabalho, não apresentando objetivos específicos da pesquisa realizada. A *Lacuna* ocorreu mais extensivamente em textos da área de Contabilidade. Quanto à *Metodologia*, os resumos apresentaram-se bastante explicativos com relação aos métodos e material utilizados, contendo ainda, por vezes, a justificativa de uso pelo de autores/referencial teórico. Sobre os *Resultados*, observamos que ocorrem nos resumos utilizando o tipo textual de descrição. Por fim, a *Conclusão* ocorreu em dois textos, sendo constituídas de breves explicações ou apresentações da importância do trabalho à área de pesquisa.

Diante disso, é possível propor que a estrutura retórica genérica e preliminar dos resumos científicos da área de Ciências Sociais Aplicadas é composta por *Contexto*, *Propósito*, *Metodologia* e *Resultado*, tendo a *Lacuna* e a *Conclusão* como macrocategorias pouco exploradas na grande área. Destaca-se também que no decorrer do trabalho e das anotações realizadas não houve sentenças que não se enquadrassem na proposta do conjunto de etiquetas utilizado.

5. Considerações finais

O estudo da estrutura retórica dos resumos das (sub)áreas de Ciências Sociais Aplicadas demonstrou-nos que há particularidades que a diferencia e a aproxima do que está proposto na literatura. Os dados obtidos a partir da anotação de cada sentença ou trecho dos textos com a identificação sintática delas constituem uma importante base para o treinamento, desenvolvimento e/ou atualização de um ambiente de auxílio à escrita acadêmica.

Em trabalhos futuros, esperamos aprofundar as análises com relação à estrutura retórica dos resumos e, em especial, apresentarmos caracterizações linguísticas de cada uma das categorias (macro e micro). Ademais, com o objetivo de propiciar uma melhor experiência de letramento acadêmico com o ambiente de auxílio à escrita, objetivamos construir um *corpus* a partir de ingressantes no Ensino Superior para detectarmos suas maiores dificuldades em aprender o GT resumo científico/acadêmico. Assim, pretendermos ter um *corpus* paralelo, em que para possíveis desvios de ordem retórica e linguística tenhamos correspondentes *gold standart*.

Referências

Antiqueira, L., Feltrim, V. D., & Nunes, M. D. G. V. (2003*). Projeto e implementação do sistema SciPo*. São Carlos, Brasil. Série de Relatórios Técnicos do Instituto de Ciências Matemáticas e de Computação (nº 223).

Bakhtin, M., & Volochinov, V. N. (2006). Marxismo e filosofia da linguagem (Vol. 7). São Paulo: Hucitec.

Castilho, R.E., Mujdricza-Maydt, E., Yimam, S. M., Hartmann, S., Gurevych, I., Frank, A., Biemann, C. (2016). A web-based tool for the integrated annotation of semantic and syntactic structures. Em *Proceedings of the workshop on language technology resources and tools for digital humanities* (LT4DH) (pp. 76-84).

Feltrim, V. D., Pelizzoni, J. M., Teufel, S., Nunes, M. D. G. V., & Aluísio, S. M. (2004). Applying argumentative zoning in an automatic critiquer of academic writing. Em *Brazilian Symposium on Artificial Intelligence* (pp. 214-223). Springer, Berlin, Heidelberg.

Feltrim, V.D. (2004). *Uma abordagem baseada em corpus e em sistemas de crítica para a construção de ambientes web de auxílio à escrita acadêmica em português*. Universidade de São Paulo, São Carlos, Brasil. Tese de Doutorado.

Hovy, E., & Lavid, J. (2010). Towards a 'science'of *corpus* annotation: a new methodological challenge for corpus linguistics. *International journal of translation*, 22(1), 13-36.

Iriguti, A. H., & Feltrim, V. D. (2019). Avaliando atributos para a classificação de estrutura retórica em resumos científicos. *Linguamática, 11*(1), pp.41-53.

Marcuschi, L. A. (2002). Gêneros textuais: definição e funcionalidade. *Gêneros textuais e ensino*, 2, pp.19-36.

Pustejovsky, J., & Stubbs, A. (2012). *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. O'Reilly Media.

Sardinha, T. B. (2004). *Linguística de corpus*. Barueri/SP: Manole Ltda.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., & Tsujii, J. I. (2012). BRAT: a web-based tool for NLP-assisted text annotation. Em *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 102-107).

Teufel, S. & Marc, M. (2002). Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational Linguistics 28*(4). 409–445.

Vieira, F. E., & Faraco, C. A. (2019). *Escrever na universidade: fundamentos*. São Paulo: Parábola.

Proceedings of Symposium in Information and Human Language Technology.
Online event, MG, Brazil, November 29th–December 3rd, 2021.

Avaliação de *parsers* na detecção de relações essenciais do modelo *Universal Dependencies* para o português

Luana Balador Belisário, Thiago Alexandre Salgueiro Pardo

Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo
São Carlos, Brasil

Resumo. *Este artigo descreve o estudo do desempenho de dois parsers conhecidos para o português com base nas diretrizes do modelo internacional "Universal Dependencies". Visando mapear o estado da arte na área, os parsers foram avaliados com relação à detecção de algumas relações essenciais do modelo que indicam os argumentos principais dos verbos. Mostramos que o parser UDPipe se destaca entre os parsers avaliados, mas que ainda há muito a avançar na área.*

1. Introdução

O modelo *Universal Dependencies*[1] (UD) (Nivre, 2015; Nivre et al., 2020) é uma proposta internacional para anotação "universal" morfossintática e sintática (incluindo características morfológicas, classes gramaticais e dependências sintáticas) de sentenças em diferentes idiomas. A iniciativa já conta com mais de 300 contribuidores, produzindo quase 200 *treebanks* anotados com as diretivas definidas do modelo para mais de 100 idiomas. A Figura 1 ilustra a anotação de uma sentença em português (reproduzida de Rademaker et al., 2017, p. 200). Pode-se ver a sentença original com suas palavras conectadas por relações de dependência sintática acima, assim como os lemas e as etiquetas morfossintáticas abaixo.

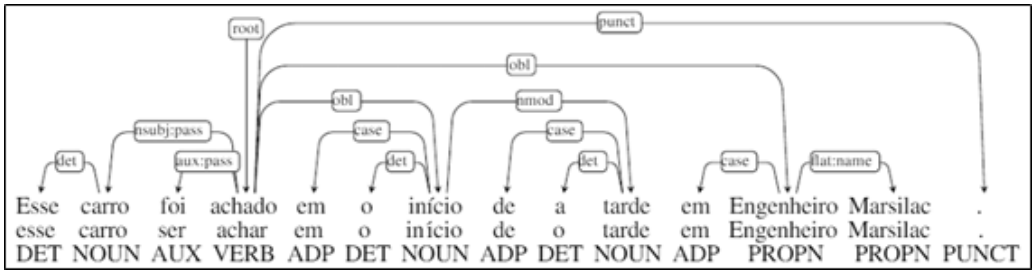


Figura 1. Exemplo de sentença em português anotada segundo o modelo UD.

Em função da grande adesão ao modelo UD e sua utilidade para o desenvolvimento de aplicações de Processamento de Linguagem Natural (PLN), *taggers* e *parsers* com base em UD têm sido criados para diversas línguas. Para o português, há alguns *parsers* que se destacam pelo uso na comunidade de pesquisa e que são objeto de análise neste trabalho, em

[1] https://universaldependencies.org/

especial, o UDPipe (Straka, 2018) e o PassPort (Zilio et al., 2018). Em seus trabalhos originais, os autores relatam valores de desempenho geral na ordem de 85 a 87% para anotação de relações sintáticas.

O objetivo deste trabalho é verificar o desempenho dos dois *parsers* citados de forma mais pontual, calculando precisão, cobertura e medida-f para algumas relações ditas essenciais (*core*) que denotam os argumentos centrais dos verbos, a saber: 'nsubj' (*nominal subject*), 'obj' (*object*) e 'iobj' (*indirect object*). Inspirada por outras iniciativas (por exemplo, Collovini et al., 2018, e Gonçalves et al., 2020), essa proposta faz parte de um esforço de mensurar de forma mais concreta os pontos fortes e fracos dos sistemas, visando fornecer subsídios para futuras pesquisas na área.

A seguir, descrevemos brevemente a metodologia adotada, sendo que os resultados obtidos são sintetizados na Seção 3. A Seção 4 apresenta algumas considerações finais.

2. Metodologia

2.1. O córpus de teste

O córpus Bosque foi anotado sintaticamente segundo as diretrizes da UD (como relatam Rademaker et al., 2017) por um grupo de pesquisadores da área e é utilizado nesse artigo como córpus de referência (*gold standard*), para avaliar a acurácia dos *parsers* testados. O córpus é composto por 9.364 sentenças e 210.957 tokens.

2.2. Um novo tokenizador

Além da versão padrão de tokenização disponibilizada com cada *parser*, também se utilizou um novo tokenizador desenvolvido no âmbito deste trabalho, mais alinhado com as diretrizes da UD, visando-se avaliar seu impacto na tarefa. O novo tokenizador, chamado LBTokenizer[2], pode ser utilizado sozinho ou integrado ao UDPipe.

2.3. A ferramenta de avaliação

O Conllu-File-Comparator[3] é um software desenvolvido na linguagem Python para essa pesquisa que compara as ocorrências das relações 'nsubj', 'obj' e 'iobj' de um arquivo com sentenças anotadas automaticamente pelos *parsers* com suas versões de referência, sendo que as anotações devem estar no formato CoNLL-U, amplamente adotado na área. Esse formato consiste em um conjunto de informações tabeladas, em que as palavras de uma sentença estão nas linhas e cada coluna armazena um tipo diferente de informação sobre as palavras.

Para cada sentença do arquivo de referência, o software contabiliza o número de cada relação essencial presente na sentença e calcula as medidas de precisão, cobertura e medida-f (assim como seus desvios padrões) para cada uma delas. Por exemplo, vamos calcular a precisão, cobertura e medida-f para uma sentença de teste com cinco tokens cujas relações essenciais estão representadas na Figura 1 e a sentença de referência com sua estrutura de relações representada na Figura 2 (note que, para simplificar a demonstração, omitimos as relações não essenciais e trocamos os tokens por números).

[2] Disponível em https://github.com/Lubelisa/LBTokenizer-UDPipe
[3] Disponível em https://github.com/Lubelisa/Conllu-File-Comparator

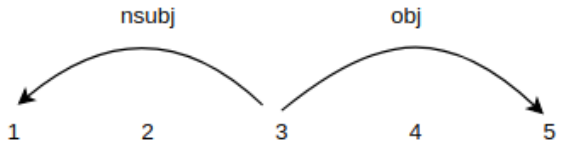


Figura 1. Sentença de teste, podendo ser anotada pelo UDPipe ou pelo PassPort

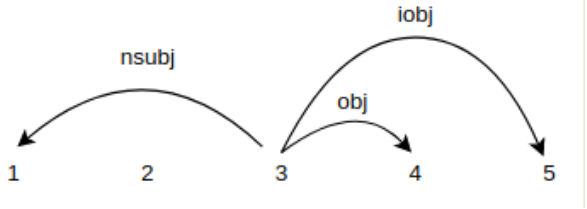


Figura 2. Sentença de referência, com anotação feita/revisada por um especialista

Na sentença de referência, há um caso de cada relação essencial e, na sentença de teste, um caso de relação 'nsubj' e um caso de 'obj'. Para calcular a precisão, precisa-se contabilizar a porcentagem de relações da sentença de teste que estão de acordo com o previsto na sentença de referência. Por exemplo, para calcular a precisão da relação 'nsubj' na sentença de teste, é necessário utilizar a fórmula a seguir.

$$Precisão_{nsubj} = \frac{número\ de\ relações\ 'nsubj'\ na\ sentença\ de\ teste\ que\ estão\ de\ acordo\ com\ a\ referência}{número\ de\ relações\ 'nsubj'\ na\ sentença\ de\ teste}$$

Assim, para a sentença de teste da Figura 1, o valor da precisão para a relação 'nsubj' é dada por $Precisão_{nsubj} = \frac{1}{1} = 1\ ou\ 100\%$. A precisão para a relação 'obj' é zero, pois, apesar de haver na sentença de teste um caso de 'obj', esse caso não é entre os mesmos tokens da sentença de referência e na mesma ordem (do token 3 para o 5).

No cálculo da cobertura, o denominador da fórmula verifica o número de relações da sentença de referência, ou seja, divide-se o número de cada relação essencial da sentença de teste que esteja de acordo com a sentença de referência pelo número de relações essenciais na sentença de referência. Logo, para a relação 'nsubj', a fórmula é:

$$Cobertura_{nsubj} = \frac{número\ de\ relações\ 'nsubj'\ na\ sentença\ de\ teste\ que\ estão\ de\ acordo\ com\ a\ referência}{número\ de\ relações\ 'nsubj'\ na\ sentença\ de\ referência}$$

Para a sentença de teste da Figura 1, o valor da cobertura para a relação 'nsubj' se dá por $Cobertura_{nsubj} = \frac{1}{1} = 1\ ou\ 100\%$. Caso houvesse dois casos de relação 'nsubj' na sentença de referência, o valor da cobertura para a relação seria de 0,5 ou 50%, portanto.

Por fim, a medida-f é uma combinação da precisão e da cobertura de cada relação, com o objetivo de se ter uma métrica única de avaliação. Foi utilizada a fórmula a seguir para calcular a medida-f de cada relação essencial:

$$Medida - f_{relação} = \frac{2{*}Precisão_{relação}{*}Cobertura_{relação}}{Precisão_{relação}+Cobertura_{relação}}$$

A seguir, relatamos os resultados obtidos para os dois *parsers* avaliados.

3. Resultados

O procedimento para a realização dos testes com os *parsers* foi passar por eles as sentenças de teste do córpus Bosque e comparar o arquivo de saída - que contém as sentenças anotadas automaticamente - com o arquivo de referência do Bosque que foi revisado por especialistas. Essa comparação e o cálculo dos resultados foi feito pelo software Conllu-File-Comparator e foram realizados três testes com os *parsers*: (1) utilizando o UDPipe para tokenizar e anotar as sentenças, (2) utilizando o LBTokenizer para tokenizar e o UDPipe apenas para anotar e (3) utilizando o PassPort para tokenizar e anotar as sentenças (pois a ferramenta foi desenvolvida para realizar as etapas em conjunto e não foi possível isolá-las). Os resultados médios dos testes (medidas e desvios padrões - DP) são mostrados na Tabela 1.

Tabela 1. Resultados dos *parsers*

	UDPipe			***LBTokenizer + UDPipe***			***PassPort***		
	nsubj	**obj**	**iobj**	**nsubj**	**obj**	**iobj**	**nsubj**	**obj**	**iobj**
Precisão	0,82	0,85	0,73	0,48	0,54	0,33	0,35	0,41	0,37
Cobertura	0,80	0,76	0,29	0,46	0,48	0,10	0,66	0,47	0,33
Medida-f	0,81	0,80	0,41	0,47	0,50	0,15	0,46	0,44	0,35
DP Precisão	0,33	0,30	0,42	0,46	0,45	0,47	0,13	0,37	0,17
DP Cobertura	0,36	0,37	0,44	0,46	0,45	0,28	0,21	0,41	0,35

Analisando-se as tabelas, conclui-se que o UDPipe sozinho obteve o melhor resultado. Para as relações 'nsubj' e 'obj', o UDPipe atingiu 80% de medida-f, mas ainda abaixo dos 87% relatados no artigo original (que engloba a avaliação de todas as relações). Há, portanto, uma grande margem para melhoria do sistema, o que se torna muito importante quando se considera que a análise produzida pelo *parser* é a entrada para outros processos em aplicações de PLN. Um erro nessa etapa pode ser propagado em outras, podendo impactar significativamente os resultados almejados. Também chama a atenção o baixo desempenho do PassPort para as três relações essenciais, apesar de, na maioria dos casos, esse sistema apresentar os menores desvios padrões para as medidas. É interessante notar que todos os sistemas avaliados têm desempenho menor para a relação 'iobj', que é sabidamente um dos casos mais desafiadores na anotação linguística com o modelo UD.

Nota-se que o uso do LBTokenizer degradou os resultados do UDPipe. Isso ocorreu porque o Bosque não segue algumas diretrizes mais atuais da UD, e usar o LBTokenizer gerou sequências de palavras diferentes das utilizadas para o treinamento do UDPipe para o português.

4. Considerações finais

Esse artigo apresenta um esforço em detalhar o desempenho de alguns *parsers* baseados no modelo UD para o português. Os resultados mostram que ainda é necessário avançar nessa frente. Trabalhos futuros incluem a avaliação de outras relações de dependência sintática e também o teste de outros sistemas, como o UDify (Kondratyuk and Straka, 2019).

Esse trabalho faz parte do projeto maior POeTiSA (*POrtuguese processing - Towards Syntactic Analysis and parsing*). Mais detalhes podem ser encontrados no portal web do projeto, em https://sites.google.com/icmc.usp.br/poetisa.

Agradecimentos

Os autores agradecem ao Centro de Inteligência Artificial da USP (C4AI - https://c4ai.inova.usp.br/), que conta com o apoio da IBM e da FAPESP (#2019/07665-4), e à Universidade de São Paulo pelo suporte financeiro.

Referências

Collovini, S.; Santos, H.D.P.; Lima, T.; Fonseca, E.; Pereira, B.; Souza, M.; Moraes, S.; Vieira, R. (2018). Cross-Framework Evaluation for Portuguese POS Taggers and Parsers. 19th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing).

Gonçalves, M.; Coheur, L.; Baptista, J.; Mineiro, A. (2020). Avaliação de recursos computacionais para o português. LinguaMÁTICA, Vol. 12, N. 2, pp. 51-68.

Kondratyuk, D. and Straka, M. (2019). 75 Languages, 1 Model: Parsing Universal Dependencies Universally. In the Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing, pp. 2779-2795.

Nivre, J. (2015). Towards a Universal Grammar for Natural Language Processing. In the Proceedings of the 16th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), pp. 3-16.

Nivre, J.; Marneffe, M-C.; Ginter, F.; Hajič, J.; Manning, C.D.; Pyysalo, S.; Schuster, S.; Tyers, F.; Zeman, D. (2020). Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In the Proceedings of the 12nd International Conference on Language Resources and Evaluation (LREC), pp. 4034-4043.

Rademaker, A.; Chalub, F.; Real, L.; Freitas, C.; Bick, E.; Paiva, V. (2017). Universal Dependencies for Portuguese. In the Proceedings of the 4th International Conference on Dependency Linguistics (Depling), pp. 197-206.

Straka, M. (2018). UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In the Proceedings of the CoNLL Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 197-207.

Zilio, L.; Wilkens, R.; Fairon, C. (2018). PassPort: A Dependency Parsing Model for Portuguese. In the Proceedings of the 13rd International Conference on the Computational Processing of Portuguese (PROPOR), pp. 479-489.

Proceedings of Symposium in Information and Human Language Technology. Online event, MG, Brazil, November 29th–December 3rd, 2021.

Utilizando Pistas Linguística para Detectar Conteúdo Enganoso em Português

Rodrigo F. Rodrigues [1], **Larissa A. de Freitas** [2]

[1]Centro de Desenvolvimento Tecnológico (CDTec)
Universidade Federal de Pelotas (UFPel) – Pelotas, RS – Brasil

{rfrodrigues,larissa}@inf.ufpel.edu.br

Abstract. *Greater access to internet-connected cell phones and the popularization of social networks have led to a significant increase in the generation and sharing of false news. Studies available in the literature, based on linguistic clues, retrieving authors of misleading content exhibits different verbal and nonverbal behavior than authors of true content. Thus, this article presents the LC-Tool tool, which extracts 29 linguistic clues from texts. Still, we tested the tool in three corpus about deceptive content available on the Internet. Finally, we realized that some linguistic clues could be extensive for the Portuguese language (e.g.: avg number of verbs and avg pausality). In other linguistic clues, they need to be validated, as they are affected by the context and domain of the messages.*

Resumo. *O maior acesso a celulares conectados à internet e a popularização das redes sociais levaram a um aumento significativo na geração e no compartilhamento de notícias falsas. Estudos disponíveis na literatura, baseados em pistas linguísticas, sugerem que os autores de conteúdo enganoso exibem comportamento verbal e não verbal diferente dos autores de conteúdo verdadeiro. Desta forma, neste artigo apresentamos a ferramenta LC-Tool, a qual extrai 29 pistas linguísticas de textos. Ainda, testamos a ferramenta em três corpus sobre conteúdo enganoso disponíveis na Internet. Por fim, percebemos que algumas pistas linguísticas podem ser extensíveis para o idioma português (por exemplo: média do número de verbos e média de pausalidade) e que em outras pistas linguísticas precisam ser validadas, pois são afetadas pelo contexto e domínio das mensagens.*

1. Introdução

O volume de informações geradas a cada minuto é enorme, nas redes sociais online (RSO), como Facebook, Twitter e Whatsapp. Uma vez que, facilitam o compartilhamento rápido de informações (Zhou and Zhang 2008). Desta maneira, surge um grande problema, a verificação da veracidade dos conteúdos compartilhados. Assim sendo, destaca-se uma área do Processamento de Língua Natural (PLN), chamada detecção de conteúdo enganoso, que pode ser realizada através do uso de pistas linguísticas.

As pistas linguísticas podem ser verbais ou não verbais (Zhou and Zhang 2008). Em que o primeiro enfoca como o engano é transmitido em uma linguagem natural e o segundo sobre o que é transmitido.

Muitos dos trabalhos de detecção de conteúdo enganoso disponíveis na literatura estão restritos ao idioma em Inglês (Zhou et al. 2003; Zhou et al. 2004; Zhou and Zhang 2008) e outros idiomas, como o Chinês (Zhou and Sung 2008) e o Russo (Litvinova et al. 2017). Para o idioma Português, existe uma escassez de conjuntos de dados rotulados sobre conteúdo enganoso. Posto isto, o presente artigo apresenta a implementação de uma ferramenta que extrai pistas linguísticas de três corpus rotulado sobre conteúdo enganoso para o idioma Português do Brasil.

O restante deste artigo está organizado da seguinte forma. Na seção 2, revisamos brevemente os trabalhos relacionados. Na seção 3, introduzimos a ferramenta proposta. Na seção 4, apresentamos a metodologia. Ne seção 5, apresentamos os resultados. Por último, na seção 6, concluímos este artigo.

2. Trabalhos Relacionados

O trabalho de (Zhou and Zhang 2008), resume os principais comportamentos de autores de conteúdo enganosos de acordo com construtores linguísticos.

Em (Fuller et al. 2006), as pistas linguísticas disponíveis nas ferramentas A99A e LIWC são avaliadas. Ainda, neste trabalho, os autores verificam que algumas pistas línguísticas, tais como: #p13, #p8, #p9 e #p14 têm diferenças significativas em mensagens sobre conteúdo enganoso em comparação com mensagens sobre conteúdo verdadeiro.

Em (Zhou et al. 2003), as pistas linguísticas são classificadas em 8 construtores linguísticos. São eles: quantidade, complexidade, não imediatismo, expressividade, afeto, especificidade, diversidade e informalidade.

Tabela 1. Resumo das pistas linguísticas.

Construtor	Pista linguística	Comportamento esperado em conteúdo enganoso	Trabalhos relacionados
Quantidade	Avg number of sentences per text - #p1 Avg number of verbs - #p2	+	(Zhou and Zhang 2008) (Fuller et al. 2006) (Zhou et al. 2003)
Complexidade	Avg size of words - #p3 Avg pausality - #p4 Avg number of sentences (in words) - #p5	-	(Zhou and Zhang 2008) (Fuller et al. 2006) (Zhou et al. 2003)
Não imediatismo	Avg number of modal verbs - #p6 Avg self reference (1st person pronoun) - #p7 Avg group reference (2nd person pronoun) - #p8 Avg another reference per text (3rd person pronoun) - #p9	+ - + +	(Zhou and Zhang 2008) (Fuller et al. 2006) (Zhou et al. 2003)
Expressividade	Avg emotiveness - #p10	+	(Zhou and Zhang 2008) (Fuller et al. 2006) (Zhou et al. 2003)
Afeto	Positive affect - #p11 Negative affect - #p12	+	(Zhou and Zhang 2008) (Fuller et al. 2006) (Zhou et al. 2003)
Especificidade	Avg spatiotemporal words - #p13	-	(Zhou and Zhang 2008) (Fuller et al. 2006) (Zhou et al. 2003)
Diversidade	Avg lexical diversity - #p14 Avg number of types per text - #p15	-	(Zhou and Zhang 2008) (Fuller et al. 2006) (Zhou et al. 2003)
Informalidade	Avg misspelled words - #p16	+	(Zhou and Zhang 2008) (Fuller et al. 2006) (Zhou et al. 2003)
Punctuation Cue	Average number of exclamation marks #p17	+	(Fernandez and Devaraj 2019)

A Tabela 1 contextualiza os construtores linguísticos nos trabalhos relaciona-

dos. O símbolo (+) indica que o comportamento da pista linguística é mais atenuado em conteúdo enganoso, enquanto o símbolo (-) indica que o comportamento da pista linguística é menos atenuado em conteúdo enganoso.

3. Ferramenta Proposta

A ferramenta LC-Tool[1] implementa um total de 29 pistas linguísticas, que podem ser aplicadas a corpus sobre conteúdo enganoso.

Na primeira etapa, é obtido o conjunto de dados a serem processados. Em seguida, na segunda etapa, é realizado o cálculo das metainformações (tokens, caracteres, sinais de pontuação e outros). Por fim, na terceira etapa, são calculadas as pistas linguísticas.

Para o desenvolvimento da ferramenta, usamos: (i) as tags *Universal POS tags*[2], com o propósito de identificar a marcação de partes da fala; (ii) o LeIA (Léxico para Inferência Adaptada)[3], com a intenção de realizar analise de sentimento no domínio de sentença; (iii) freeoffice pt-BR[4] em conjunto com o spaCy[5], com o objetivo de encontrar erros ortográficos.

4. Metodologia

Para realização deste trabalho, o primeiro passo foi buscar conjuntos de dados rotulados com conteúdo enganoso para a língua portuguesa do Brasil. Em segundo lugar, era necessário um estudo sobre detecção de conteúdo enganoso com base em abordagens que fazem uso de pistas linguísticas. Em terceiro lugar, implementamos uma ferramenta que extrai pistas linguísticas de corpus rotulados com conteúdo enganoso, usando a linguagem de programação Python. Por fim, cada conjunto de dados foi utilizado na ferramenta para que as pistas linguísticas fossem calculadas. Assim, é possível avaliar os resultados indicados pelas pistas linguísticas em cada conjunto de dados.

Em nossos experimentos, utilizamos três conjuntos de dados: anônimo-1[6], FakeTweetBr[7] e Fake.br-Corpus[8]. O anônimo-1 contém notícias sobre a cura do COVID-19. O FakeTweetBr é um corpus de notícias falsas do Twitter. O Fake.br-Corpus contém notícias classificadas em seis grandes categorias (política, TV e celebridades, sociedade e notícias diárias, ciência e tecnologia, economia, religião).

5. Análise dos Resultados

As pistas linguísticas sugerem a probabilidade de que o transmissor esteja tentando enganar. Sendo assim, as pistas linguísticas estão inseridas em várias unidades de texto, incluindo palavras, frases, sentenças ou mensagens. A Tabela 2 apresenta os resultados obtidos nos três conjuntos de dados.

[1]https://github.com/pseudorfrodrigues/LinguisticCluesTool
[2]https://universaldependencies.org/docs/u/pos/
[3]https://github.com/Deceptive-Content-Utilities/LeIA
[4]https://www.freeoffice.com/pt/baixar/dicionarios
[5]https://spacy.io/
[6]https://wp.ufpel.edu.br/midiars/datasets/
[7]https://github.com/prc992/FakeTweet.Br
[8]https://github.com/roneysco/Fake.br-Corpus

Tabela 2. Resultados obtidos nos três datasets.

Pista linguística	anônimo-1: outro	anônimo-1: desmentido	anôniomo-1: desinformação	Fake.br: True	Fake.br: False	FakeTweetBr: True	FakeTweetBr: False
#p1	2.09	2.32	**2.75**	58.30	12.00	3.73	**4.07**
#p2	10.47	12.05	**16.58**	11.88	**12.97**	10.98	**13.00**
#p3	4.80	5.05	**4.52**	4.88	4.90	6.20	**5.54**
#p4	0.96	0.93	**0.77**	2.81	**2.64**	1.39	**1.26**
#p5	8.79	8.50	**8.27**	18.92	**15.05**	7.72	**6.99**
#p6	2.78	2.17	**3.02**	2.64	**2.95**	3.40	**3.42**
#p7	0.00∼	0.00∼	0.50	0.21	**0.19**	0.12	**0.09**
#p8	0.00∼	0.00∼	0.00∼	0.16	0.13	0.00∼	**0.09**
#p9	5.98	9.88	**12.06**	8.18	8.11	5.67	**7.25**
#p10	0.50	0.44	**0.50**	0.45	0.43	0.36	0.35
#p11	0.22	0.26	0.12	0.31	0.30	0.26	**0.28**
#p12	0.22	0.58	**0.75**	0.69	0.69	0.55	**0.60**
#p13	9.19	11.81	**11.56**	10.69	11.87	16.68	**15.41**
#p14	0.68	0.70	**0.69**	0.34	0.52	0.70	0.72
#p15	12.43	13.79	15.62	379.79	**93.26**	20.15	20.60
#p16	0.71	2.94	2.20	1.39	**1.83**	3.82	3.16
#p17	0.21	0.48	**1.01**	0.03	**0.31**	0.68	**2.20**

O presente trabalho buscou descobrir as possibilidades do uso de pistas linguísticas para distinguir conteúdos verdadeiros de enganosos no idioma português brasileiro. Encontramos 6 pistas linguísticas (#p2, #p4, #p5, #p6, #p12 e 17) que obtiveram o comportamento sugerido pelos trabalhos relacionados, para os três conjuntos de dados, 5 pistas linguísticas em dois conjuntos de dados (#p1, #p3, #p9, #p7 e #p13) e 6 pistas linguísticas em apenas um conjunto de dados (#p16, #p14, #p15, #p11, #p10 e #p8).

Nos estudos de (Zhou and Zhang 2008), escritores de conteúdo enganoso possuem maior afeto positivo e negativo, porém, a teoria da perspectiva auto-apresentacional diz que eles são menos positivos e agradáveis, o que foi verificado neste trabalho.

Esperava-se que a emotividade fosse maior em conteúdos enganaosos, mas observamos que o número de adjetivos e advérbios é menor em quantidade nas informações enganosas, sendo essas duas informações importantes para o cálculo da emotividade. Dessa forma, obtivemos menos emotividade no Fake.Br e no FakeTweetBr.

Nos trabalhos de (Zhou et al. 2003; Zhou et al. 2004; Zhou and Zhang 2008), foi possível observar maior informalidade em conteúdos enganosos. Porém, é importante ressaltar que os conjuntos de dados usados por estes autores são balanceados. Em nossos experimentos, o único conjunto de dados que resultou em mais informalidade foi o Fake.br, que foi o único conjunto de dados balanceado utilizado, os demais eram desbalanceados. O que pode ter contribuído com a baixa acertividade dessa pista linguística.

6. Conclusões

Este trabalho apresentou a ferramenta LC-Tool que extrai pistas linguísticas de corpus rotulados com conteúdo enganoso. Encontramos 6 pistas que alcançaram o comportamento esperado nos conjuntos de dados anônimo-1, Fake.br e FakeTweetBr.

O estudo da detecção de conteúdo enganoso é importante para a comunidade acadêmica e a sociedade em geral. Nessa perspectiva, estamos trabalhando para que em trabalhos futuros possamos realizar experimentos utilizando técnicas de aprendizado profundo. Para isso, pretendemos aplicar o modelo BERTimbau em conjuntos de dados

com conteúdo enganoso escritos em língua portuguesa.

Referências

Fernandez, A. C. and Devaraj, M. (2019). Computing the linguistic-based cues of fake news in the philippines towards its detection. pages 1–9.

Fuller, C., Biros, D., Twitchell, D., Burgoon, J., and Adkins, M. (2006). An analysis of text-based deception detection tools. volume 6, page 418.

Litvinova, O., Seredin, P., Litvinova, T., and Lyell, J. (2017). Deception detection in Russian texts. In *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 43–52, Valencia, Spain. Association for Computational Linguistics.

Zhou, L., Burgoon, J., and Douglas, T. (2004). A comparison of classification methods for predicting deception in computer-mediated communication. 20(4):139–165.

Zhou, L., Burgoon, J., Twitchel, D., Quin, T., and Jay, N. (2003). An exploratory study into deception detection in text-based computer-mediated communication. 20(4):1–10.

Zhou, L. and Sung, Y.-w. (2008). Cues to deception in online chinese groups. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pages 146–153.

Zhou, L. and Zhang, D. (2008). Following linguistic footprints: Automatic deception detection in online communication. *Commun. ACM*, 51(9):119–122.