Submission date: 20/09/2021
Resubmission date: 30/01/2022
Camera ready submission date: 23/04/2022

1st round notification: 27/12/2021
2nd round notification: 25/03/2022
Available online: 05/08/2022
Publication date: 05/08/2022

Section: extended version
Extended version of selected papers from the XVI Brazilian Symposium on Information Systems

Comparison of natural language processing techniques in social bot detection on Twitter during Brazilian presidential elections

Bianca Lima Santos[1], **Gabriel Estavaringo Ferreira**[1], **Marcelo Torres do Ó**[1], **Rafael Rodrigues Braz**[1], **Luciano Antonio Digiampietri**[1]

[1]University of São Paulo – (USP)
São Paulo, SP – Brazil

{bianca_lima, estavaringo, marcelo.torres.o, rafael.braz, digiampietri}@usp.br

Abstract. *Currently, there are thousands of social bots acting on different online social networks. Identifying them automatically is a computational challenge. This work uses different natural language processing methods to extract features from tweets collected during the 2018 Brazilian presidential election period in order to make the bot detection process more precise. The developed solution uses artificial intelligence techniques, combining feature selection and classification algorithms. The authors obtained the best results through a union of all the extracted features using the Random Forest classifier, achieving an precision of 0.86 for the bot class and AUC of 0.86.*

Keywords. *Bot detection, Social networks, Twitter, Elections, Natural language processing, Machine learning.*

1. Introduction

Internet access has been growing in Brazil. In 2018, 79,1% of residences had access to the internet. Increasingly more people can access online banking services, news, and social networking services [IBGE 2018]. Social networking services, also known as online social networks, are typically used for entertainment, to keep in touch with friends, and for sharing photos, updates, and news. They can also be used to exercise democracy through debates. Twitter is one of the most popular of such services and works as a personal micro blog where each user can post messages of up to 280 characters (tweets), view messages posted by other users, express reactions and share messages from other users with their network of followers.

Online social networks proved to be very effective in influencing people and malicious users have used this ability to manipulate public opinion and spread disinformation [Soroush Vosoughi 2018]. With the malicious use of social bots, that is, computational algorithms that simulate a user using the network, it is possible to exploit this potential by automatically spreading messages across the network [Ferrara et al. 2016]. Bots can

Cite as: Santos, B. K., Ferreira, G. E., Torres do O, M., Braz, Rafael Rodrigues & Digiampietri, L. (2022). Comparison of naturallanguage processing techniques in social bot detection on Twitter during Brazilian presidential elections (Comparação de técnicas de processamento de linguagem natural na detecção de bot sociais no Twitter durante as eleições presidenciais brasileiras). iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems), 15(1), 12:1-12:22. DOI: 10.5753/isys.2022.2225.

be beneficial, such as weather forecast bots or chatbots on websites. But the presence of malicious bots on social networks is undesirable and harmful. By studying the use of bots in the 2016 US elections, [Ferrara et al. 2016] concluded that their presence could negatively affect the democratic debate. Social bots can post or share many messages to support or attack politicians or political subjects, and they can spread disinformation to achieve their objectives. Moreover, it is not difficult to set up a bot operating on social media due to the variety of information available and bots ready to be customized. A new trend known as Bot-As-A-Service (BaaS) provides bots in an easy-to-use way as a paid service [Ferrara et al. 2016].

It's estimated that many bots were used on Twitter during the 2018 Brazilian presidential elections to influence and direct political debates within the service. The robots' behavior would have been directed at defending certain candidates and demoralizing candidates by spreading defamatory and fake news [Leu et al. 2019]. Identifying and actively acting to inhibit the presence of bots that propagate fake news on social networking services and actively participate in political campaigns is directly related to promoting digital democracy.

The research presented in this work aimed to use the content of messages posted by Twitter users during the 2018 presidential elections in Brazil to identify potential bot users. For this identification, the authors used six different techniques of textual representations individually and in combination, evaluated by six different classifiers, to understand if there is a difference between the texts produced by bots and humans in such a way that it is possible to distinguish them. In addition, features related to the user profile and number of publications were added [Santos et al. 2020] to analyze which is the best set of available features for detecting bots.

Natural Language Processing (NLP) is an interdisciplinary field based on theories from linguistics, computer science, and artificial intelligence that aims to give computers the ability to process natural languages (such as human languages) [Jurafsky and Martin 2009]. The text analysis can be organized in several stages: lexical analysis, syntactic analysis, semantic analysis, and pragmatic analysis [Dale et al. 2000]. The lexical analysis focus on the identification of meaningful lexemes. The syntactic analysis complements the lexical analysis by assessing the relationship among the lexemes. Semantic analysis aims to identify the meaning of a word or a sentence. Pragmatic analysis tries to identify the goals or intentions of the speaker [Dale et al. 2000, Jurafsky and Martin 2009]. While each stage of analysis allows the identification of new information, bringing the interpretation closer to the real meaning and intention of the speaker, there are several challenges in automating this analysis. The first decades of text processing (until the early 2000s) were limited to syntactic analysis, the current decade and the next ones will stand out for semantic analysis, and the last decades of the present century will stand out for pragmatic analysis [Cambria and White 2014].

This work corresponds to an extension of our previous work [Ferreira et al. 2021]. This extension includes alternative ways to represent and extract information from the content of the tweets posted during the 2018 Brazilian presidential elections. In particu-

lar, we include the use of text transformers [Vaswani et al. 2017]. This paper is organized as follows: Section 2 presents some related works, Section 3 describes the materials and methods used for the experiments carried out, Section 4 contains the results and discussions, and Section 5 presents the conclusions and future work.

2. Related Work

There are four main approaches commonly used to detect social bots: network-based detection, crowd-sourcing, feature-based detection, and a combination of the former [Ferrara et al. 2016]. A well-known bot detection tool is *Botometer*, formerly known as *BotOrNot?* [Davis et al. 2016], which implements a detection algorithm based on targeted features to detect social bots' suspicious behavior.

There is a lot of discussion about the use of bots in political contexts. The [Bessi and Ferrara 2016] study used the *BotOrNot?* tool to detect bots on Twitter and then examine their impact on the 2016 US elections. In one of the experiments, sentiment analysis was used to compare the positivity of tweets produced by humans and bots about the two major candidates. On the other hand, the work of [Hurtado et al. 2019] uses network analysis and temporal analysis techniques to detect the activity of bots in political discussions on the Reddit social network.

Only few papers use natural language processing (NLP) for the extraction of user's features, and there are even fewer papers that use a Brazilian Portuguese corpus. [Chu et al. 2012] used CRM114 to classify user's publications as spam or not. This information was combined with the user's features and the time entropy of publications to classify users among bot, human, or cyborg (a cyborg profile corresponds to an account controlled by both a human and a bot). In contrast, [Mohammad et al. 2019] proposed a model using convolutional neural networks to detect bots considering the textual content of a single publication.

The paper of [Dickerson et al. 2014] uses a set of user's features based on the sentiments of posts as a way to improve accuracy in bot detection on Twitter. [Cai et al. 2017] uses temporal information of the tweets and LSTM convolutional neural networks to classify a user as bot or human.

[Dukić et al. 2020] used Bidirectional Encoder Representations from Transformers (BERT) to verify if a tweet was written by a bot or a human. For each token, they proposed a representation composed of three parts: a 768-dimensional BERT embedding, a 300-dimensional emoji embedding, and tweet-specific features. The former embedding is a summary of the semantic context of the tweet and was produced from a pre-trained BERT without fine-tuning. The classification was performed with two classifiers: Feed-Forward Deep Neural Network and Logistic Regression.

In [de Morais and Digiampietri 2021] a systematic review is carried out on the detection of bots in social networking services, and brought as a conclusion the need for the emergence of new techniques for detecting bots, since the identified techniques could not guarantee the efficiency of the long-term approach.

Two papers are the basis of the current work. In [Santos et al. 2020], we deve-

loped a social bot detection approach considering only 34 features derived from user characteristics such as post count, number of followers, and post frequency. This paper used a training set of 800 Twitter profiles manually labeled as bot or human to train five different classifiers using 10-folds cross-validation. In [Ferreira et al. 2021], we included some features extracted from textual representations of user posts to improve the classification. The present work is an extension of this previous work that used NLP, by including features extracted from a pre-trained BERT model, the state of the art in a textual representation.

3. Materials and Methods

This section describes the dataset, problem modeling, and strategies used.

3.1. Dataset

The dataset used is a sample of 800 different users and 111,530 posts in Brazilian Portuguese published during the second round of the 2018 Brazilian presidential elections. This dataset was made available by [Leu et al. 2019] and manually labeled by [Santos et al. 2020] into bot and human classes.

3.2. Tokenization

Tokenization is the process of separating a document into smaller units called tokens. The tokens can be words, characters, or even a string of bytes. During tokenization, it was necessary to preserve links, hashtags (word or agglutination of words preceded by a # used to identify subjects or topics), mentions (word or agglutination of words preceded by an @ used to quote names of other network users) and emoticons such as “=)”, “:-)”, and “:(”.

3.3. Term frequency–inverse document frequency

Term frequency–inverse document frequency (TF-IDF) is a statistical measure to reveal the relevance of a term in a document. It is used in conjunction with tokenization to generate numerical features from a tokenized document.

The term frequency (TF) is given by the number of times the term i appears in the document j ($n_{i,j}$) divided by the total number of terms in the document j, and is given by equation 1.

$$\mathrm{tf}(i,j) = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

On the other hand, the inverse document frequency (IDF) calculates the weight of terms that appear in the corpus, considering that less frequent terms have greater weight, and is given by equation 2.

$$\mathrm{idf}(i) = \log\left(\frac{N}{n_i}\right) \tag{2}$$

Where N is the total number of words in all documents in the corpus and n_i is the number of times the term i occurs. Thus, the TF-IDF is given by the product of the TF and the IDF:

$$\text{tf-idf}(i,j) = \frac{n_{i,j}}{\sum_k n_{k,j}} \cdot \log\left(\frac{N}{n_i}\right) \tag{3}$$

To perform the TF-IDF calculation, it was used the implementation available in *scikit-learn*, a *machine learning* library in the Python programming language.

3.4. Textual representations

Six textual representation techniques were used to produce numeric features from the text of user's tweets: character n-grams, word n-grams, part-of-speech tagging, sentiment analysis, Word2Vec, and BERT embeddings. For both sentiment analysis and BERT embeddings, the approach used was to keep one document per tweet. For all other representations, all tweets from each user were merged into a single document.

3.4.1. Character n-grams

An n-gram is a contiguous sequence of text elements. These elements can be characters, words, bytes, syllables, phonemes, or any other element relevant to the text. Splitting a document into n-grams is a commonly used technique for natural language processing and three n-gram strategies were employed in this work: characters, words, and part-of-speech.

From the tweets of each user, sub sequences of adjunct elements (words or characters) were produced. These sequences have length n, where n is a positive integer. For characters n-grams, each character of the sentence is an element that, according to the given n, clumps with the next $n-1$ elements. For example, in the sentence: "good morning" for n=2 the sentence results in the following list of 2-grams: "go", "oo", "od", "d ", " m", "mo", "or", "rn", "ni", "in" and "ng". In this work, character n-grams of size ranging from 1 (unigram) to 5 were used as features. The pipeline used for preprocessing posts using character n-grams is presented in Figure 1.

3.4.2. Word n-grams

The word n-gram approach benefits from the previous explanation, differing only from the subdivision strategy, which in this case is word-based instead of character-based. During the processing of these n-grams, punctuation marks and *stop-words* (common words in the language that are considered without semantic value) were discarded. For example, in the sentence: "Today is too hot" for $n = 2$ you would get the following 2-grams: "Today is", "is too" and "too hot". This work used word n-grams with sizes ranging from 1 to 5. The pipeline used for preprocessing the word n-gram is presented in Figure 2.

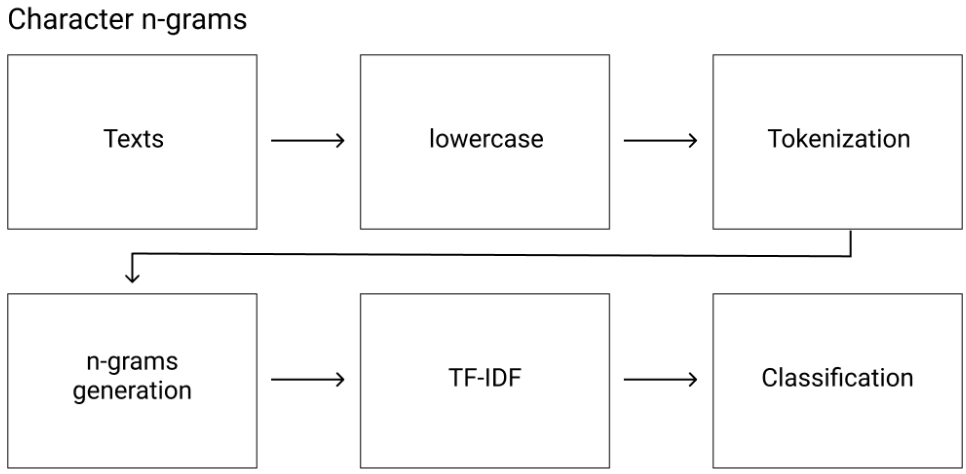

Figura 1. Character n-grams preprocessing architecture [Ferreira et al. 2021]

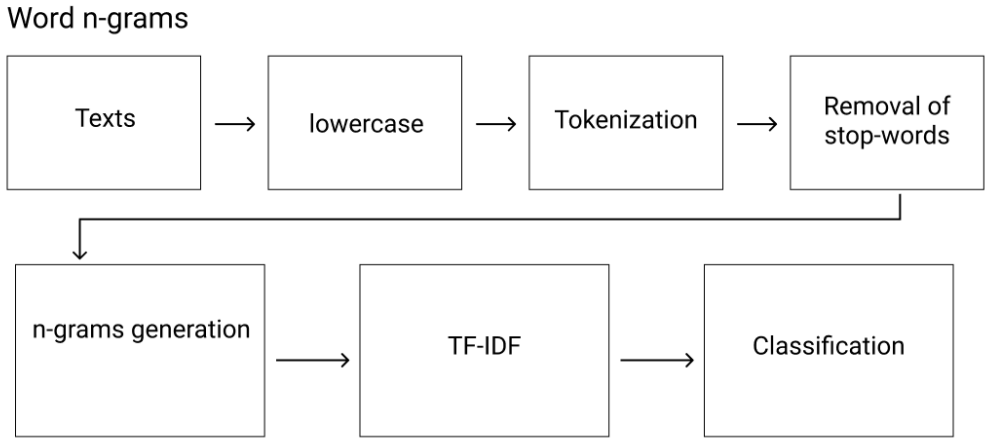

Figura 2. Word n-grams preprocessing architecture [Ferreira et al. 2021]

3.4.3. Part-of-speech Tagging

Part of speech is an expression to refer to a classification of words according to their common morphological, semantic or syntactic characteristics [Meisel 1990]. Therefore, *part-of-speech tagging* or *POS tagging* refers to the process of labeling each word in a text with its part-of-speech. These part-of-speeches can then replace the words in the original text, which is seen as a sequence of tags. This representation is useful to analyze the syntactic construction of texts produced by a user and not their meaning, as the original words are lost.

Thus, in this work, the use of *POS tagging* aims to verify whether it is possible to differentiate human users and bots from the analyzed dataset based on the grammatical structures used by each one. The tagging was done using the *tagger* provided by NILC (*Interinstitutional Center for Computational Linguistics*) in [Fonseca and Rosa 2013],

considered the state of the art for part-of-speech tagging in Portuguese. In addition to 30 *tags* present in the *tagger*, new tags were added that, despite not representing the traditional Portuguese language, bring additional information about the language used on the internet, namely: EMAIL, WEBLINK, HASHTAG, and MENTION. From the sequences of tags, n-grams with sizes from 1 to 5 were extracted and each n-gram was then analyzed as a feature of the dataset, calculating the relative frequencies with *TF-IDF* in an analogous way to n-grams of words and characters.

3.4.4. Sentiment Analysis

Sentiment analysis is a natural language processing technique used to identify and extract opinions or emotions [Souza et al. 2011]. Different techniques can be used to analyze sentiments in a text, including the use of lexicons and the use of machine learning algorithms.

There are lexicons available in Portuguese, such as the OpLexicon [Souza et al. 2011], which has a corpus of 7,077 words and is built by combining different lexicons, showing improvement when comparing the results with the uncombined versions. [Sousa 2016] proposes a semantic polarity classifier using the SentiWordNet lexicon [Esuli and Sebastiani 2006] translated from English, obtaining satisfactory results when comparing the results using a resource already in Brazilian Portuguese. Other interesting approach is proposed by [Gilbert 2014], which implements a sentiment analyzer focused on texts expressed in social media. Based on some rules and using a lexicon, this implementation obtained a F-measure $= 0.96$ when classifying the sentiment of 4,200 tweets into positive, neutral and negative classes.

The sentiment analyzer used in this work is a Portuguese adaptation of [Gilbert 2014]. Proposed in [Almeida 2018], it has lexicons and dictionaries translated from English, but preserves the original implementation. To generate the features in the sentiment analysis approach, the tweets were analyzed and, after removing accents and replacing the emojis by their textual description, the polarity score for each of the tweets was calculated, consisting of four values, the positive percentage, negative percentage, neutral percentage and the general sentiment value normalized between -1 and 1. With these values, a total of nine features were generated per user, which were used as input for the classifiers. The normalized general sentiment value was used to obtain the amount of tweets for each of the sentiments. Based on the implementation of [Gilbert 2014], the thresholds used to classify the tweet as positive, negative or neutral were -0.05 and 0.05. In addition, the sum and average of positive, negative and neutral percentages of each tweet were calculated.

3.4.5. Word2Vec

One of the ways to analyze semantics in NLP is based on the hypothesis that words that occur in similar contexts tend to have similar meanings. This principle is used in word embedding representations [Jurafsky and Martin 2009].

Word embedding is the representation of words as vectors of real numbers capable of capturing syntactic, semantics and/or morphology of words, depending on the chosen embedding algorithm [Hartmann et al. 2017]. One of such algorithms is Word2Vec, proposed in [Mikolov et al. 2013], capable of generating vectors that capture the semantics of words. This algorithm can generate embeddings in two different ways: Skip-Gram, in which, given a word, the algorithm tries to predict the neighboring words and Continuous Bag-of-Words (CBOW), where the middle word given the neighbors is predicted.

There are embeddings in Portuguese trained in a large corpus and made available by [Hartmann et al. 2017], but for the present work 300 dimensions vectors were trained based on the tweets of about 1,200,000 users contained in the dataset provided by [Leu et al. 2019] and that are not among the 800 labeled users. As the tweets are within the context of the 2018 presidential elections, it is believed that they can generate more significant vectors as the texts contain words and expressions specific to the 2018 political debate. For example, in this context the word "lula" is more likely to refer to the politician than to the mollusk ("lula" also means "squid" in Portuguese).

For the vector generation, the Gensim library implementation of Word2Vec in Skip-gram mode was used, while for the generation of tokens, four different preprocessing approaches were tested, which are better described in Table 1. Each of the approaches was used to process the tweets of the 800 labeled users and the texts used to generate the vectors.

Tabela 1. Pre-processing approaches

Name	**Description**
P_{RAW}	TweetTokenizer application only
P_{SIMPLE}	P_{RAW} plus all lowercase and replace urls, email, numbers by "URL", "EMAIL" and "0"
P_{NILC}	P_{SIMPLE} plus the other preprocessing options provided by [Hartmann et al. 2017]
P_{NILC_EMOJI}	P_{NILC} plus the replacement of emojis by their description

Two types of features were produced, as described in Table 2. For each type, three values were defined for n; $n \in \{10, 20, 50\}$ for Window Vector features; and $n \in \{100, 500, 1000\}$ for the TF-IDF Embedding Vectorizer features. Each one of the attributes was tested in combination with the four preprocessing approaches described.

Tabela 2. Attribute types

Name	Description
Window Vector	The sequence of n vectors with the highest TF-IDF mean within the document
TfIdf Embedding Vectorizer	The mean of the n vectors with the highest TF-IDF weighted by the TF-IDF

3.4.6. Bidirectional Encoder Representations from Transformers - BERT

Models based on Transformers are helpful in natural language processing problems. These models can perform tasks such as translations, extract information, quiz games, text generation or classification. They are, typically, composed of two modules: an encoder and a decoder, as we see in Figure 3. The former is responsible for processing the input sequence and transforming it into context. The second is responsible for producing the output sequence. In a translation problem, for example, the input is the text in a source language that will be processed, generating language-free contexts, such as symbols that together represent that inserted sentence. Then this context is passed to the decoder, which transforms the context into the text of the target language, thus carrying out the translation process.

Bidirectional Encoder Representations from Transformers, also known as BERT, is a language representation model that uses encoder modules from transformers. It has the advantage that it can be pre-trained and fine-tuned for a variety of tasks [Devlin et al. 2018]. In BERT, each word is relevant to the context and its contextualization is built bidirectionally. It is achieved considering the context from the right and the left of the word.

Because of the BERT capability of transferring learning, there are different pre-trained models in a variety of languages. In this work, a pre-trained model in the Portuguese language was used. Proposed in [Souza et al. 2020], BERTimbau provides two models pre-trained using a large Portuguese dataset composed of Brazilian web pages: BERTimbau base and BERTimbau large. The first has 12 layers and 110 million parameters, while the second has 24 layers and 335 million parameters.

For this paper, we use the python implementation of BERTimbau large available on Huggingface Transformers library with the PyTorch library. The model has an input limit of 512 tokens and works with 768-dimensional embeddings. The tokenization was carried out with the BERT tokenizer provided by the same library. For the problem addressed in the scope of this work, that is, detection of bots, there are two possible approaches for using a pre-trained BERT model to perform the classification: fine-tuning or using BERT embeddings as input for external classifiers.

In **BERT fine-tuning**, an classification head (usually a feed-forward network fol-

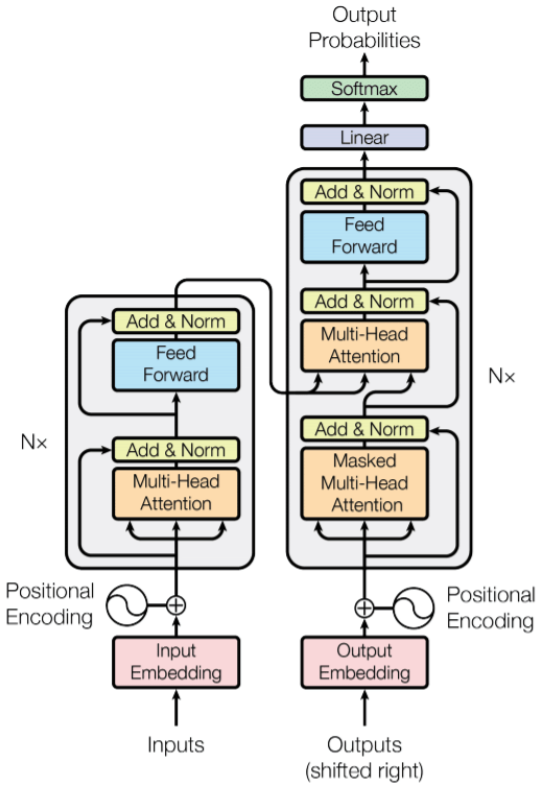

Figura 3. The encoder-decoder model architecture of a transformer [Vaswani et al. 2017]

lowed by a soft-max function) is coupled to a pre-trained BERT. This model then goes through further training iterations so that the network becomes specialized for the task. Fine-tuning requires a much smaller amount of data for training as it builds on top of an already trained BERT with millions of data. For this work, when using fine-tuning with BERT, it was not able to detect any bot, classifying all users as human. This is due to the low number of users labeled in the dataset. Furthermore, this approach does not allow the incorporation of additional features independent of BERT in the classification, making the fine-tuning approach just a little comparable to the others adopted in this work.

The other possibility is to use the **BERT embeddings** as input to classifiers, without classifying directly. Among the outputs of the last hidden layer of BERT, there is an embedding for a special token called [CLS] that contains a summary of the semantic content of the input and is typically used in classification tasks.

In this article, the strategy of extracting features from BERT embeddings was adopted. Thus, the text of each tweet from each user was treated in such a way as to remove emojis, URLs, mentions, hashtags, and emails and then tokenized. The BERT limit of 512 input tokens is not a problem when dealing with individual tweets, but it would be a problem when trying to process all tweets as a single document. Then, for each user, each tokenized tweet was processed by BERT, producing the embeddings. However, users have a different amount of tweets and the selected classifiers work with a uniform input. Therefore, to produce a fixed amount of features to feed the classifiers, for each

user an average embedding was extracted from the embedding [CLS] of each tweet, thus producing 768 features per user.

3.5. Classification Strategies

3.5.1. Classifiers

The classifiers used in this work were selected among the most common in the literature and after testing them with the sets of generated features. They were also selected trying to mix simple models and more modern and complex models. Namely, the selected classifiers were: Decision Tree, Naive Bayes, Multilayer Perceptron, Random Forest, and Support Vector Machine. For all classifiers, the results were calculated using 10-fold cross-validation.

Decision Tree is an algorithm that builds a path in a graph based on attributes that most contribute to the model. At each node a path is chosen based on the feature that proves to be the most advantageous for the model. The chosen implementation for this work is based on the CART (Classification and Regression Tree) algorithm, which is very similar to the C4.5 algorithm. It works by building a binary tree taking the feature and the threshold value that produce the greatest information gain to the model.

Naive Bayes is a probabilistic model based on Bayes Theorem that assumes variables are independent for classification purposes. However, this independence between variables is not always true for many datasets [Chen et al. 2020]. In this paper, the chosen implementation was Gaussian Naive Bayes, which assumes that the likelihood of the features is Gaussian. The Multinomial variant was tested, but the former produced better results.

Multilayer Perceptron (MLP) is a neural network of elements called perceptrons arranged in layers. There are three kinds of layers: the input layer, the hidden layer, and the output layer. The MLP model can have different configurations varying the number of hidden layers and the number of perceptrons in each one of them. Other hyperparameters include the learning rate and the activation functions. The model presented in this work uses an architecture of 100, 250, and 50 perceptrons in three hidden layers.

Random Forest is an ensemble classifier, that is, a classifier consisting of a set of tree classifiers constructed with random perturbations. These trees are trained from sampling with replacement from a random subset of all features. The Random Forest classifies by voting from the classification provided by all the trees in the ensemble.

Support Vector Machine is a model that performs a classification identifying the best hyperplane which separates the instances based on data transformed with a technique called “kernel trick”.

3.5.2. Feature Set

In order to identify which preprocessing strategy produces the features that contribute the most on identifying bots, every classifier was trained using different sets of features

produced by each of the preprocessing stategies. Therefore, the results are presented separately according to the three strategies used for obtaining features listed below:

- Strategy A: each classifier was trained with only the features extracted by each specific textual representation described in Section 3.4.
- Strategy B: each classifier was trained with features from each specific textual representation plus the original dataset features from [Santos et al. 2020].
- Strategy C: each classifier was trained with the combined features extracted from all textual representations described in this work plus the features from [Santos et al. 2020].

3.5.3. Feature Selection

One of the factors that influence the success of machine learning applications is the amount of information provided to the model. Irrelevant or redundant information may negatively affect the classification [Hall 2000].

Since several features were obtained from the preprocessing strategies described in Section 3.4, it became necessary to use selectors to remove redundant or irrelevant features. For the word n-grams, character n-grams, and POS tagging approaches, the KBest selector implemented by the scikit-learn library with the F statistic of ANOVA score was used.

Additionally, an own implementation in Python language of the Correlation-based Feature Selector (CFS) selector, based on the definition of [Hall 2000], was used for the Sentiment Analysis approach. The implementation of CFS uses the local search algorithm Hill climbing and works searching for the local optima feature set. When there are so many features, as is the case of embeddings, the CFS takes a long time to execute.

For embeddings generated from this dataset, the execution of the KBest selector with Mutual Information (MI) was beneficial and was faster than applying CFS. But using the CFS over the selected features from Kbest was quicker and was further helpful. Thus, both word2vec and BERT embeddings approaches used two selectors together: the KBest selector with the Mutual Information (MI) score function, followed by the CFS.

3.6. Evaluation Strategy

To evaluate the results, the following metrics obtained from classifications were used: precision, recall, F measure, and area under the ROC curve (AUC). The accuracy, given by $\frac{TP+TN}{TT}$, is not a very relevant measure for this classification problem, because the data has many more records labeled as *human* than as *bot*. The measures are defined below:

$$Precision = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1} = \frac{2 * precision * recall}{precision + recall}$$

Considering *bot* as the positive class and *human* as the negative class, TP (true positives) is the number of bots correctly classified as bots; FN (false negatives) is the number of bots wrongly classified as humans, FP (false positives) the number of humans wrongly classified as bots; TN (true positives) is the number of humans correctly classified as humans; and TT the total number of labeled users in the dataset.

4. Results and discussion

This section presents the results obtained, a brief discussion and a comparison among the different techniques used. The results obtained with the word n-grams and character n-grams techniques are presented in Table 3. Table 4 presents the results achieved through the POS Tagging and Word2Vec techniques, and Table 5 presents the results achieved by using sentiment analysis and BERT techniques. Finally, the results found using the combination of all features are summarized in Table 6.

4.1. Sentiment Analysis

When analyzing the results obtained with the feature set A, that is, using only the sentiment analysis features, the Random Forest classifier obtained the highest F1-score, with the value equal to 0.84. If we compare the precision, the algorithms SVM, Decision Tree and Random Forest achieved similar results, with 0.83, 0.82 and 0.82, respectively (see Table 5).

On average, the classifiers achieved results similar to those reported in [Santos et al. 2020]. However, if we compare only the bot class, we can see that the results were worse when using only the features generated through the sentiment analysis, this may be an indication that bots express sentiments in a similar way to humans in posts on Twitter. With feature set B, Naive Bayes and Random Forest algorithms were the best, achieving 0.85 precision and F1-score. The results reported weren't better than the results obtained in [Santos et al. 2020].

4.2. Character n-grams

Character n-grams performed better with feature set A, especially when considering the bot class, according to Table 3. The best result was obtained with the Naive Bayes classifier, in which the precision for the bot class was 0.81, a very expressive value despite the low recall. For feature set B, which adds the features obtained in [Santos et al. 2020] to the features generated in the context of this work, the best results were achieved by Naive Bayes and Random Forest classifiers.

4.3. Word n-grams

Classification using the feature set generated with the word n-grams technique achieved a performance that can be considered inferior to that when using character n-grams. The best result for the bot class was obtained by using the Naive Bayes classifier. The result obtained by Random Forest for feature set A was also satisfactory, reaching AUC equal to 0.80. When using the SVM classifier with feature set B, the performance of character and word n-grams was equivalent (see Table 3).

Tabela 3. Results - Word n-grams and character n-grams [Ferreira et al. 2021]

			word n-grams				character n-grams			
ft.	**classifier**	**class**	**pre.**	**rec.**	**F1**	**AUC**	**pre.**	**rec.**	**F1**	**AUC**
A	NB	bot	0.28	0.65	0.39	0.81	0.81	0.32	0.46	0.74
		human	0.94	0.78	0.86		0.92	0.99	0.95	
		mean	0.87	0.77	0.80		0.91	0.91	0.90	
	SVM	bot	0.07	0.37	0.12	0.55	0.46	0.48	0.47	0.73
		human	0.82	0.38	0.52		0.93	0.93	0.93	
		mean	0.73	0.38	0.47		0.88	0.87	0.88	
	DT	bot	0.25	0.27	0.26	0.58	0.58	0.33	0.42	0.57
		human	0.90	0.90	0.90		0.92	0.97	0.94	
		mean	0.83	0.82	0.82		0.88	0.90	0.88	
	RF	bot	0.64	0.10	0.17	0.80	0.70	0.32	0.44	0.71
		human	0.89	0.99	0.94		0.92	0.98	0.95	
		mean	0.86	0.89	0.85		0.89	0.91	0.89	
	MLP	bot	0.50	0.01	0.02	0.78	0.00	0.00	0.00	0.73
		human	0.88	1.00	0.94		0.88	1.00	0.94	
		mean	0.84	0.88	0.83		0.78	0.88	0.83	
B	NB	bot	0.31	0.55	0.39	0.80	0.33	0.35	0.34	0.81
		human	0.93	0.84	0.88		0.91	0.91	0.91	
		mean	0.86	0.80	0.83		0.85	0.84	0.84	
	SVM	bot	0.17	0.57	0.26	0.61	0.17	0.58	0.26	0.62
		human	0.92	0.62	0.74		0.92	0.62	0.74	
		mean	0.83	0.62	0.69		0.83	0.62	0.69	
	DT	bot	0.30	0.30	0.30	0.60	0.35	0.37	0.36	0.63
		human	0.91	0.91	0.91		0.92	0.91	0.91	
		mean	0.84	0.84	0.84		0.85	0.85	0.85	
	RF	bot	0.27	0.04	0.07	0.79	0.69	0.24	0.35	0.79
		human	0.89	0.98	0.93		0.91	0.99	0.95	
		mean	0.81	0.88	0.83		0.88	0.90	0.88	
	MLP	bot	0.37	0.15	0.21	0.63	0.23	0.08	0.11	0.66
		human	0.90	0.97	0.93		0.89	0.97	0.93	
		mean	0.83	0.87	0.85		0.81	0.86	0.83	

By analyzing the results for both N-gram techniques, we can see that there are specific language markers for bots that allow to identify them, however, it is not possible to assume that bots simply always repeat the same phrases or words, given the lower performance for the word n-grams.

4.4. POS Tagging

Among the feature set generated with POS tagging, features such as “WEBLINK MENTION V ADJ PU” (weblink, mention, verb, adjective, punctuation) and “WEBLINK WEBLINK MENTION PROPESS” (weblink, weblink, mention, personal-pronoun) stand out, as they represent the importance of analyzing the language of the social network.

Tabela 4. Results - POS Tagging and Word2Vec [Ferreira et al. 2021]

			POS Tagging				Word embedding			
ft.	**classifier**	**class**	**pre.**	**rec.**	**F1**	**AUC**	**pre.**	**rec.**	**F1**	**AUC**
A	NB	bot	0.76	0.58	0.66	0.80	0.48	0.35	0.41	0.79
		human	0.95	0.98	0.96		0.92	0.95	0.93	
		mean	0.92	0.93	0.93		0.87	0.88	0.87	
	SVM	bot	0.46	0.55	0.50	0.78	0.26	0.80	0.39	0.81
		human	0.94	0.92	0.93		0.96	0.70	0.81	
		mean	0.88	0.87	0.88		0.88	0.71	0.76	
	DT	bot	0.61	0.32	0.42	0.51	0.17	0.18	0.18	0.53
		human	0.92	0.97	0.94		0.89	0.88	0.89	
		mean	0.88	0.90	0.88		0.81	0.80	0.80	
	RF	bot	0.80	0.35	0.49	0.78	0.50	0.02	0.04	0.76
		human	0.92	0.99	0.95		0.89	1.00	0.94	
		mean	0.91	0.92	0.90		0.84	0.88	0.83	
	MLP	bot	0.00	0.00	0.00	0.79	0.49	0.23	0.31	0.80
		human	0.88	1.00	0.94		0.90	0.97	0.94	
		mean	0.78	0.88	0.83		0.86	0.88	0.83	
B	NB	bot	0.36	0.40	0.38	0.83	0.52	0.55	0.53	0.86
		human	0.92	0.91	0.91		0.94	0.93	0.94	
		mean	0.85	0.85	0.85		0.89	0.89	0.89	
	SVM	bot	0.17	0.58	0.26	0.61	0.14	0.57	0.22	0.57
		human	0.92	0.62	0.74		0.90	0.53	0.67	
		mean	0.83	0.62	0.69		0.81	0.54	0.62	
	DT	bot	0.37	0.35	0.36	0.63	0.20	0.18	0.19	0.54
		human	0.92	0.92	0.92		0.89	0.90	0.90	
		mean	0.85	0.86	0.85		0.81	0.82	0.81	
	RF	bot	0.83	0.26	0.39	0.85	0.50	0.04	0.08	0.86
		human	0.91	0.99	0.95		0.89	0.99	0.94	
		mean	0.90	0.91	0.89		0.84	0.88	0.84	
	MLP	bot	0.30	0.14	0.19	0.58	0.57	0.42	0.48	0.86
		human	0.89	0.96	0.92		0.93	0.96	0.94	
		mean	0.82	0.86	0.84		0.89	0.90	0.84	

When analyzing the results obtained with feature set A, the Naive Bayes algorithm obtained the best results, with F1 score = 0.93, precision = 0.92 and area under the ROC curve equal to 0.80. The Random Forest algorithm obtained similar results, reaching F1 = 0.9 and precision = 0.91 (see Table 4).

Regarding the results found with feature set B, the best result was obtained by the Random Forest algorithm, with precision equal to 0.9 and measure F = 0.89. The results with the RF algorithm are particularly interesting because it is possible to reach a precision greater than 0.8 for the bot class, even if the recall is much lower when comparing to NB algorithm. In general, the results of the POS tagging technique indicate that some

Tabela 5. Results - Sentiment analysis and BERT

			Sentiment analysis				BERT			
ft.	**classifier**	**class**	**pre.**	**rec.**	**F1**	**AUC**	**pre.**	**rec.**	**F1**	**AUC**
A	NB	bot	0.00	0.00	0.00	0.61	0.20	0.85	0.32	0.79
		human	0.88	1.00	0.94		0.96	0.55	0.70	
		mean	0.78	0.88	0.83		0.88	0.58	0.65	
	SVM	bot	0.17	0.51	0.25	0.59	0.15	0.58	0.24	0.40
		human	0.91	0.67	0.78		0.91	0.58	0.71	
		mean	0.83	0.66	0.71		0.82	0.58	0.66	
	DT	bot	0.22	0.22	0.22	0.55	0.25	0.31	0.28	0.60
		human	0.90	0.90	0.90		0.91	0.88	0.89	
		mean	0.82	0.82	0.82		0.83	0.81	0.82	
	RF	bot	0.29	0.10	0.15	0.78	0.57	0.04	0.08	0.73
		human	0.89	0.97	0.93		0.89	1.00	0.94	
		mean	0.82	0.87	0.84		0.85	0.89	0.84	
	MLP	bot	0.11	0.10	0.10	0.50	0.17	0.02	0.04	0.69
		human	0.88	0.90	0.89		0.88	0.99	0.93	
		mean	0.79	0.81	0.80		0.80	0.87	0.83	
B	NB	bot	0.34	0.41	0.37	0.80	0.21	0.84	0.34	0.82
		human	0.92	0.90	0.91		0.97	0.59	0.73	
		mean	0.85	0.84	0.85		0.88	0.62	0.69	
	SVM	bot	0.17	0.58	0.27	0.63	0.16	0.59	0.25	0.38
		human	0.92	0.63	0.75		0.92	0.60	0.72	
		mean	0.83	0.63	0.69		0.83	0.60	0.67	
	DT	bot	0.31	0.34	0.32	0.62	0.23	0.23	0.23	0.56
		human	0.91	0.90	0.91		0.90	0.90	0.90	
		mean	0.84	0.83	0.84		0.82	0.82	0.82	
	RF	bot	0.50	0.13	0.21	0.81	0.43	0.10	0.16	0.79
		human	0.90	0.98	0.94		0.89	0.98	0.94	
		mean	0.85	0.88	0.85		0.84	0.88	0.85	
	MLP	bot	0.26	0.20	0.23	0.70	0.36	0.09	0.14	0.75
		human	0.90	0.93	0.91		0.89	0.98	0.93	
		mean	0.82	0.84	0.83		0.83	0.88	0.84	

grammatical structures can be good markers to identify bots in the analyzed data.

4.5. Word Embedding

For the feature set A the best result was found with 44 Window Vector attributes of $n = 50$ considering P_{NILC_EMOJI}, according to Table 4. For the feature set selected from attributes $A \cup B$, they were composed of 44 Window Vector attributes of $n = 50$ created with P_{NILC} pre-processing and the attributes: geo_enabled, number_screenname_ratio, tweets_tag_per_active _interval, max_tweets_3h and tweets_per_day_diff.

The Decision Tree, Random Forest and SVM models did not show satisfactory performance in the bots classification task in any of the two sets. However, the MLP and

Tabela 6. Results - Features combined

ft.	classifier	class	pre.	rec.	F1	AUC
C	NB	bot	0.36	0.56	0.44	0.83
		human	0.94	0.87	0.90	
		mean	0.87	0.83	0.85	
	SVM	bot	0.16	0.57	0.25	0.62
		human	0.54	0.59	0.50	
		mean	0.83	0.61	0.68	
	DT	bot	0.34	0.33	0.34	0.62
		human	0.91	0.92	0.91	
		mean	0.85	0.85	0.85	
	RF	bot	0.86	0.13	0.22	0.86
		human	0.90	1.00	0.94	
		mean	0.89	0.90	0.86	
	MLP	bot	0.54	0.38	0.44	0.73
		human	0.92	0.96	0.94	
		mean	0.88	0.89	0.88	

Naive Bayes algorithms presented satisfactory performance.

Compared to the work of [Santos et al. 2020], the MLP and Naive Bayes classifiers showed better results, especially when combined with the previous features. The Random Forest, on the other hand, had worse results. Thus, the use of Word2Vec brought positive gains to the MLP and Naive Bayes algorithms.

4.6. BERT Embeddings

Despite being a pre-trained complex model with millions of parameters, the classification results using only features produced by BERT, shown in Table 5, did not stand out in relation to other textual representations, having a similar performance to the classification using features produced with word2vec, except for some using SVM. For this classifier, BERT produced poor results, which indicates that the embeddings produced by BERT cannot be easily separated in a linear fashion. An advantage of BERT over word2vec was the use of a pre-trained model in Portuguese, without the need to train a specific model for the problem.

The Naive Bayes classifier was the most effective in detecting bots, obtaining a high recall, despite the low accuracy. On the other hand, tree-based models, in particular Random Forest, have a low recall for bots, but they do it with much greater precision, avoiding false positives. By combining the features extracted from the BERT embeddings with user features in set B, it was possible to observe an improvement for the Random Forest and MLP classifiers, considerably increasing their ability to recall bots.

4.7. Combined Features

By combining the features of all six pre-processing techniques addressed in the context of this work, it was observed that, in general, the performance obtained by the classifiers

was superior to the use of each technique individually, shown in Table 6. When compared to the work which this one extends [Ferreira et al. 2021], the performance improvement of the MLP classifier is noticeable, which could not detect any bots, but now has a recall of 0.38 and accuracy of 0.54. It is noteworthy that MLP needs a lot of calibration of hyperparameters to reach its maximum performance, a task not performed in the scope of this work, since the intention was to compare the different algorithms and feature sets in a balanced way.

On average, the best ranking algorithm for this approach was Random Forest, which had an accuracy of 0.89, recall of 0.90, F1-score of 0.86, and AUC of 0.86. Also, looking specifically at the bot class, this classifier achieved an precision of 0.86, being most recommended if we want to target identifying bots while minimizing the number of false positives.

The Naive Bayes classifier also achieved a good overall performance, with an F1-score equal to 0.85 and an area under the ROC curve equal to 0.83. With a recall of 0.56 for the bot class, this was the algorithm with the greatest ability to detect bots, but the accuracy of 0.36 makes it not recommended in situations where it is important to avoid false positives.

5. Conclusion

Among the six natural language processing techniques used, the POS tagging approach had the best results when comparing the average across classes. Using only the features generated in the scope of this work, the results obtained by it were superior to all other techniques. Considering the classifiers, the Naive Bayes and Random Forest algorithms stood out, obtaining good results regardless of the technique used.

Analyzing each feature set extracted by different textual representation approaches apart, the results were similar or superior to the results using just user features without natural language processing [Santos et al. 2020]. Thus, it is concluded that when we analyze only the texts published by a user, we can identify bots with an equivalent or greater precision than when looking only at the user's features. However, by combining both techniques we can identify bots with a higher precision than if we only consider the user features.

Using BERT, the state-of-the-art model in natural language processing, in general, brought results similar to other pre-processing techniques. One of the difficulties involving this technique was the limitation in the number of input tokens. Also, averaging the embeddings of each user's tweets may have generated some noise in the final embedding. Another approach that can be used in future work is the use of a recurrent neural network. Thus, the network would be feed with all the user's tweets in a sequence, preserving some of the individual information of each one.

This technique shows itself as a promising way to detect social bots that, as shown in Section 2, lacks new techniques and approaches. It can be used in the context of political debate in order to mitigate the problem of spreading fake news and hate speech, promoting digital democracy on Twitter or other social networking services.

For future work, the sentiment analysis technique could be improved to generate a different set of features. An aspect-oriented approach can be interesting for the task proposed in this work since the dataset has very diverse topics that can be used to classify bots. An interesting approach is proposed in [Dickerson et al. 2014] which, through an aspect-oriented approach, shows that the features generated using sentiment analysis increased the accuracy of the classification algorithm. In addition, other sentiment analyzers could be tested, including methods that use machine learning.

Another approach to be used is the analysis of the network of each profile, followers, and people who are followed. This analysis could produce/calculate new features to be used in classification. Finally, the techniques described could be used with a similar purpose in different social networks not addressed in the context of this work.

Referências

[Almeida 2018] Almeida, R. J. A. (2018). Leia - léxico para inferência adaptada. `https://github.com/rafjaa/LeIA`.

[Bessi and Ferrara 2016] Bessi, A. and Ferrara, E. (2016). Social bots distort the 2016 u.s. presidential election online discussion. *First Monday*, 21(11). `https://doi.org/10.5210/fm.v21i11.7090`.

[Cai et al. 2017] Cai, C., Li, L., and Zeng, D. (2017). Detecting social bots by jointly modeling deep behavior and content information. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1995–1998, New York, NY, USA. Association for Computing Machinery.

[Cambria and White 2014] Cambria, E. and White, B. (2014). Jumping nlp curves: A review of natural language processing research [review article]. *IEEE Computational Intelligence Magazine*, 9(2):48–57.

[Chen et al. 2020] Chen, S., Webb, G. I., Liu, L., and Ma, X. (2020). A novel selective naïve bayes algorithm. *Knowledge-Based Systems*, 192:105361. `https://doi.org/10.1016/j.knosys.2019.105361`.

[Chu et al. 2012] Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. (2012). Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing*, 9(6):811–824.

[Dale et al. 2000] Dale, R., Moisl, H., and Somers, H. (2000). *Handbook of Natural Language Processing*. Taylor & Francis.

[Davis et al. 2016] Davis, C. A., Varol, O., Ferrara, E., Flammini, A., and Menczer, F. (2016). Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 273–274, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

[de Morais and Digiampietri 2021] de Morais, D. M. G. and Digiampietri, L. A. (2021). Methods and challenges in social bots detection: A systematic review. In *XVII Brazilian Symposium on Information Systems*, SBSI 2021, New York, NY, USA. Association for Computing Machinery.

[Devlin et al. 2018] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[Dickerson et al. 2014] Dickerson, J. P., Kagan, V., and Subrahmanian, V. S. (2014). Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '14, page 620–627, New York, New York, USA. IEEE Press.

[Dukić et al. 2020] Dukić, D., Keča, D., and Stipić, D. (2020). Are you human? detecting bots on twitter using bert. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 631–636.

[Esuli and Sebastiani 2006] Esuli, A. and Sebastiani, F. (2006). SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 417–422, Genoa, Italy. European Language Resources Association (ELRA).

[Ferrara et al. 2016] Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. (2016). The rise of social bots. *Commun. ACM*, 59(7):96–104. https://doi.org/10.1145/2818717.

[Ferreira et al. 2021] Ferreira, G. E., Santos, B. L., do Ó, M. T., Braz, R. R., and Digiampietri, L. A. (2021). Social bots detection in brazilian presidential elections using natural language processing. In *XVII Brazilian Symposium on Information Systems*, SBSI 2021, New York, NY, USA. Association for Computing Machinery.

[Fonseca and Rosa 2013] Fonseca, E. R. and Rosa, J. L. G. (2013). Mac-morpho revisited: Towards robust part-of-speech tagging. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, Porto Alegre, Brazil. SBC.

[Gilbert 2014] Gilbert, C. H. E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, Menlo Park, California, USA. Association for the Advancement of Artificial Intelligence.

[Hall 2000] Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 359–366, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Hartmann et al. 2017] 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. https://sol.sbc.org.br/index.php/stil/article/view/4008.

[Hurtado et al. 2019] Hurtado, S., Ray, P., and Marculescu, R. (2019). Bot detection in reddit political discussion. In *Proceedings of the Fourth International Workshop on*

Social Sensing, SocialSense'19, page 30–35, New York, NY, USA. Association for Computing Machinery.

[IBGE 2018] IBGE (2018). Instituto brasileiro de geografia e estatística. acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal. In *PNAD Contínua 2018*. Available on: https://www.ibge.gov.br/estatisticas/sociais/trabalho/17270-pnad-continua.html?edicao=27138&t=resultados.

[Jurafsky and Martin 2009] Jurafsky, D. and Martin, J. H. (2009). *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, Upper Saddle River, N.J.

[Leu et al. 2019] Leu, M. D. O., Morais, D. M. G., Xavier, F., and Digiampietri, L. A. (2019). Detecção automática de bots em redes sociais: um estudo de caso no segundo turno das eleições presidenciais brasileiras de 2018. In *Revista de Sistemas de Informação da FSMA*, page 31–39, Macae, Brazil. FSMA.

[Meisel 1990] Meisel, W. S. (1990). Speech representation and speech understanding. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '90, page 423, USA. Association for Computational Linguistics.

[Mikolov et al. 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Stroudsburg, USA. Association for Computational Linguistics.

[Mohammad et al. 2019] Mohammad, S., Khan, M. U. S., Ali, M., Liu, L., Shardlow, M., and Nawaz, R. (2019). Bot detection using a single post on social media. In *2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4)*, pages 215–220, New York, New York, USA. IEEE Press.

[Santos et al. 2020] Santos, B. L., Ferreira, G. E., do Ó, M. T., Braz, R. R., and Digiampietri, L. A. (2020). Comparação de algoritmos para detecção de bots sociais nas eleições presidenciais no brasil em 2018 utilizando características do usuário. *Revista Brasileira de Computação Aplicada*, 13(1):53–64.

[Soroush Vosoughi 2018] Soroush Vosoughi, Deb Roy, S. A. (2018). The spread of true and false news online. *Science*, 359(6380):1146–1151. https://doi.org/10.1126/science.aap9559.

[Sousa 2016] Sousa, R. C. C. d. (2016). Identificando sentimentos de texto em português com o sentiwordnet traduzido. Technical report, Universidade Federal do Ceará, Campus de Quixadá, Quixadá.

[Souza et al. 2020] 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)*.

[Souza et al. 2011] Souza, M., Vieira, R., Busetti, D., Chishman, R., and Alves, I. (2011). Construction of a portuguese opinion lexicon from multiple resources. In *Proceedings*

of the 8th Brazilian Symposium in Information and Human Language Technology, pages 59–66, Porto Alegre, RS, Brazil. SBC.

[Vaswani et al. 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.