Revista Brasileira de Computação Aplicada, July, 2023

DOI: 10.5335/rbca.v15i2.14045
Vol. 15, Nº 2, pp. 1–10
Homepage: seer.upf.br/index.php/rbca/index

RBCA
ISSN 2176-6649

ORIGINAL PAPER

Authorship attribution of comments in Portuguese extracted from Reddit

Vinicius Alves Matias [1] and Luciano Antonio Digiampietri [1]

[1]School of Arts, Sciences and Humanities, University of São Paulo

*viniciusmatias@usp.br; digiampietri@usp.br

Received: 2022-11-17. **Revised:** 2023-07-16. **Accepted:** 2023-07-27.

Abstract

Internet interaction environments such as social networks transfer large-scale textual data that implicitly carry the writing styles of each network user. Given the constant and intense flow of information through information systems of this type, it is necessary to develop techniques that can distinguish a text between two candidate authors for reasons of, for example, avoiding the return of users banned from the platform. This paper addressed and evaluated different ways of performing authorship attribution through natural language processing and machine learning, based on comments in Portuguese extracted from Reddit social network. This paper aims to update the authorship attribution literature using Portuguese as the primary language given the scarcity of updated works in this language. The results of several viable methods for the task of binary authorship were exposed and evaluated in the question of feasibility according to their statistical significance, achieving two independent models in the same confidence interval that reached 0.88 of F1-score and 0.94 of AUC with extraction of textual attributes through BERTimbau embeddings and through TF-IDF of words.

Keywords: Authorship Attribution; Natural Language Processing; Machine Learning; Social Networks; Text Mining

Resumo

Os ambientes de interação da Internet, como as redes sociais, transferem dados textuais em larga escala que carregam implicitamente os estilos de escrita de cada usuário da rede. Dado o fluxo constante e intenso de dados nos sistemas de informação deste tipo, torna-se necessário desenvolver técnicas que consigam distinguir um texto entre dois candidatos a autores por motivos de, por exemplo, evitar o regresso de utilizadores banidos da plataforma. Este artigo abordou e avaliou diferentes formas de realizar atribuição de autoria por meio de processamento de linguagem natural e aprendizado de máquina, com base em comentários em português extraídos da rede social Reddit. Este artigo visou a atualizar a literatura de atribuição de autoria usando o português como língua principal, dada a escassez de trabalhos atualizados neste idioma. Os resultados de vários métodos viáveis para a tarefa de autoria binária foram expostos e avaliados de acordo com sua significância estatística e foram encontrados dois modelos independentes no mesmo intervalo de confiança que atingiu 0,88 de F1-score e 0,94 de AUC com extração de atributos textuais a partir de embeddings BERTimbau e utilizando de TF-IDF de palavras.

Palavras-Chave: Aprendizado de Máquina; Atribuição de Autoria; Mineração de texto; Processamento de linguagem natural; Redes sociais

1 Introduction

Social Networks are online environments in which users can, in a simple way, create a user account and carry out conversations with other members of the network. The facility creating an account and the attraction for more users to a social network can also be a problem for its management, given that banned users can return to the platform and contribute, for example, to the proliferation

of fake news.

Reddit is a social network recognized for the anonymity given to its users. The recommendation of posts and comments from this system is based on the number of people who liked some content versus those who explicitly reported that they did not, leading to a range of publications with sensational headlines, making Reddit a network known for spreading rumors that are sometimes nothing more than fake news.

The idea of this social network is to allow users to create and participate in communities on a specific topic, known as subreddits. Users can then post and comment on different subreddits, on topics such as politics, movies, and pets, according to the subreddit's purpose.

Given the characteristics of Reddit, where it is possible that there are many users who discuss similar subjects in specific communities, expanding the feeling of freedom of expression with anonymity, this social network (or online discussion forum) is presented as a source of interesting data for training and testing authorship attribution methods. In addition, automatic ways of identifying authors by text can be very useful in this type of network to help, for example, hold a user accountable for the continuous violation of platform rules or local legislation and identify multiple profiles of the same user.

Authorship attribution is the task of recognizing the author who wrote a text through his writing style, and to perform attribution in an automated way. Natural language processing and machine learning techniques are used. The application of such methods is varied, and so is the way of approaching problems. As an example of application, there is the detection of plagiarism, identifying which writing styles of a text are not compatible with what is expected of an author through authorial detection. Another application is the recognition of socioeconomic characteristics through the result of author characterization algorithms, which in turn recognize the patterns of different groups segmented through the characteristics studied.

Research on authorship attribution in Portuguese contains few updates regarding the computational advances made in recent years with more advanced textual classification techniques using neural networks. Still, an evaluation of the performance of different methods for classifying authors in this problem is important to identify if an technique really has a significant result, and if simpler and faster techniques can be equivalent, even facilitating the explanation of why a text has been assigned to an author.

This paper deals with the problem of binary authorship attribution - when a new text can only be classified as belonging to a finite set of authors, which, in this case, are two. Different classification and text processing methods were evaluated to quantitatively assess their differences and statistical significance.

2 Related Work

Several methods for authorship detection have been used over the years and with different purposes. This observation is verifiable by the extensive literature review produced by Swain et al. (2017). The study describes the seven subareas of the field of authorship recognition and analysis, also known as stylometry (an author's writing styles according to a linguistic bias), ranging from the previously mentioned authorship characterization, to the problem of authorship attribution - given a finite set of authors and texts written by them, identify which were the authors of texts not analyzed until then. The attribution problem is viewed as a classification problem that involves text mining and natural language processing. The review carried out identified that most of the published studies selected for review used classical machine learning techniques and a promising future for the use of neural networks of more advanced architectures. Text attribute extraction commonly took advantage of n-gram vectorization of characters and POS tagging.

Regarding the data sources, there is a field of extreme variety. Through the analysis of the article, it is possible to perceive that there is a tendency to use texts from social networks and blogs, but also applicable to divergent topics, such as the identification of the author of a source code written in C++ or Java, as well as identifying the authors of literary works based on texts from public domain books (Swain et al., 2017). A consequent problem in the area, but sometimes disregarded, is the authorship attribution of short texts (such as comments), as an example extracted from the review, there is a study that used texts from ancient Arab travelers, reaching about 80% precision for resolution of this problem.

Among the attribution methods applied to social interaction environments on the Internet, the article of Abbasi and Chen (2005) stands out for its pioneering nature. Texts in Arabic and English were captured from extremist forums to perform the attribution task using a Support Vector Machine. In the pre-processing stage, lexical analysis and word count were performed to represent the text to the classifier. Like many later works, this one focused on performing the detection only of texts with a minimum length (since there are texts that are even humanly impossible to distinguish, such as a laugh or the use of an emoji).

Twitter is a social network that also generates interest in the topic due to the ease of data extraction and the large number of interactions among users. This drew the attention of Layton et al. (2010) to test an n-gram technique taking advantage of the character delimiting of texts. Among the text character substitutions performed in the work, it is noteworthy that the removal of mentions of people from a tweet did not lead to a significant interference for the authorship classification.

Casimiro and Digiampietri (2020, 2022) published two studies on the Author Attribution problem using English texts extracted from Reddit for sets of multiple authors, reaching an accuracy of 99% for 10 authors and 70% for 100 authors. However, there is a low number of publications using texts and approaches for Portuguese, which is an open challenge given that the NLP area has made several advances in the last ten years, especially after the publication of the language representation model BERT (Devlin et al., 2019), reaching state-of-the-art for many text classification problems. A recent multi-language authorship detection work was developed

by Custódio and Paraboni (2019), using an ensemble approach of classifiers and n-grams of characters to compare their performance against an SVM (typical baseline) and analyzing different pre-processing and data representation strategies.

This paper performs an update and analysis of different approaches for authorship attribution in texts with an unrestricted amount of characters, and for this purpose, comments in Portuguese from social networks are used. The data source used was extracted from Brazilian Reddit communities described in the methodology.

3 Background

3.1 Tokenization and N-grams

Machine learning algorithms work by performing various calculations to identify a decision rule that represents the data, implying that the input to these algorithms must be numbers. Each text in a corpus (name commonly given to the set of texts) is composed of a set of tokens, which are the textual parts that together form the example of your data samples. Thus, tokens can be words, characters and all the elements that form your text individually or in sequence. One of the ways to measure the value of each token in a text is, for example, to use the number of times each token appears in the text. N-grams are used to generate attributes based on unique values or strings of characters or words (Fig. 1).

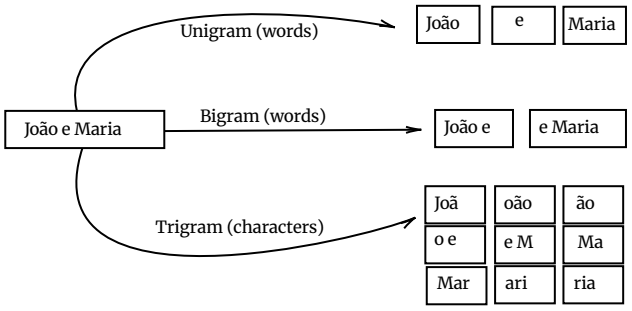

Figure 1: Diagram of conversion of the text "João e Maria" to attributes based on unigrams and bigrams of words, and trigams of characters. In character vectorization (generating feature vectors from the text) spaces participate in some 3-character tokens.

3.2 TF-IDF

Another way of weighting attributes is by looking at the frequency of a term in a text, but also at the frequency of the term in other documents - Eq. (1).

$$w_{i,j} = tf_{i,j} \log\left(\frac{N}{df_i}\right) \quad (1)$$

Since i is the actual attribute and j is the text in which this value was found, the weight $w_{i,j}$ of this attribute is calculated by multiplying the term-frequency in this document $tf_{i,j}$ by the inverse of the relative frequency of the term in the documents, that is, the total amount of N texts divided by the number of documents with the analyzed term df_i. The log is used for scaling purposes that allow the frequency of terms to have a significant impact on the TF-IDF.

3.3 Part-of-Speech Tagging

Part-of-Speech (POS) tagging is the task of assigning to which grammatical class a word belongs. Since it is not possible to map all the words of a language considering different contexts, nor keep this reference updated, when it is necessary to perform an automatic mapping from word to grammatical class it is common to use models already developed for this task. spaCy Python library (Honnibal et al., 2022) maintains and makes available a model trained with data from Wikipedia (Nothman et al., 2013) and news (Rademaker et al., 2017), both with references to the parts of speech sought. Fig. 2 exemplifies how to convert a simple sentence based on the model.

3.4 Word Embeddings and Word2vec

The use of n-grams of characters and words to extract attributes from a text is a classic approach, but it loses information about the context in which a word is used. A method to represent words in another way is through word embeddings. In this approach the vocabulary (each word of the corpus) is analyzed through a similarity process to define the representation of a word in vector format.

Word2vec uses the terms and neighborhood collected in a neural network architecture, as described in the article in which it was proposed (Mikolov et al., 2013). One of the architectures is called Continuous Bag-of-Words (CBOW), which consists of a neural network where the input is the vectors of nearby words in their n-gram word frequency representation connected to a hidden layer that will represent the vector (whose size N is required) of embeddings of a word. The hidden layer will be connected with the output layer, which will have to adjust the weights until it is able to predict the vector that represents the central word of the given context in the input.

3.5 Bidirectional Encoder Representations from Transformers

The Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) took advantage of the encoders from the transformers architecture (Vaswani et al., 2017) to develop an approach that pays attention to the context of a sentence in both directions (using all document words), going beyond approaches that typically check the context based on the left to right of a sentence.

Combining the method's architecture with a large

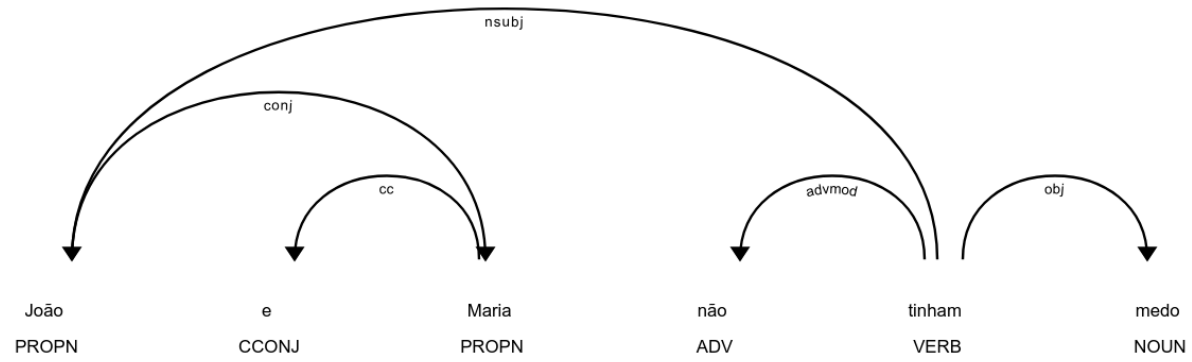

Figure 2: Syntactic tree of a sentence in Portuguese.

training corpus, BERT is able to approach the state-of-the-art for different tasks related to natural language processing, including text classification. With BERT it is possible to perform a fine-tuning with the addition of an output layer in the original model, adjusting the training with the data of a new task. It is also possible to collect the representation encoded by BERT for the tokens of a text and thus be able to use them with the same idea of word embeddings in classification models.

To improve the performance of a task that uses BERT as a reference, it is still necessary to note the language in which the classification is being performed. For Portuguese there is the pre-trained BERT known as BERTimbau (Souza et al., 2020)

3.6 Classification Models

Machine learning models can be used to employ binary classification, hence valid for binary authorship. For this, the models mentioned here receive a training set and seek to create a classification rule that represents the data seen, which can be based on probability calculations, function estimation, class separation calculations, etc. This paper uses Multinomial Naive Bayes, Logistic Regression with L1 and L2 regularization, Support Vector Machines (with linear and RBF kernels), Decision Trees, Random Forest, AdaBoost, Gradient boosting and Stacking of classifiers.

4 Methodology

4.1 Dataset

To create the dataset, the authors of the 1000 most recent posts from the *Brasil*, *brasilivre* and *BrasildoB* subreddits were collected using the Python PRAW library (Boe, 2022), whose date of collection was April 17, 2022. After identifying the top recent authors from each of these subreddits, we collected the 1000 most recent comments from these 15 authors, reaching a dataset composed of 15,000 comments, also distinguished by author and comment date.

For each of the 15 authors, duplicate comments were removed, so that no two texts are the same for one author, but it is possible to have duplicates between different authors. Comments composed of emojis or laughter are examples of texts that can be the same between authors. With this pre-processing step, there are 14,520 comments to be used. For visualization purposes, figures Fig. 3 and Fig. 4 present a sample of texts with less than 620 characters (95% of the dataset). By analyzing the figures and the measures mentioned, it can be noted that there is a predominance of comments with few characters.

4.2 Feature Extraction from Text

The following text processing was performed for the authorship attribution task, in order to present the text to the classifiers:

- **Frequency and TF-IDF using the original comments**: Unique words and strings of up to 3 words; strings of characters in the range 1-5, 4-5 and 3-8.
- **Frequency and TF-IDF using POS tagging of the comments**: Unique words and strings of up to 3 words.
- **word2vec**: The word embeddings were generated by training with the entire corpus of 14520 dataset documents, creating 100 dimensions per word.
- **BERTimbau**: Using pre-trained model embeddings.

The use of POS tagging is not intuitive to be used with embeddings, since there will be few terms that will appear many times, generating a lot of noise. BERT only accepts texts with a length less than or equal to 512 tokens, therefore, to use this model for linguistic representation, it was necessary to remove comments with more tokens than the maximum supported. It was a reduction of 967 texts, reaching 13,553 documents in the corpus. To generate the embeddings of the documents, the average of each dimension of the attributes produced by BERTimbau and word2vec was taken, with each document represented, respectively, by 100 and 768 attributes

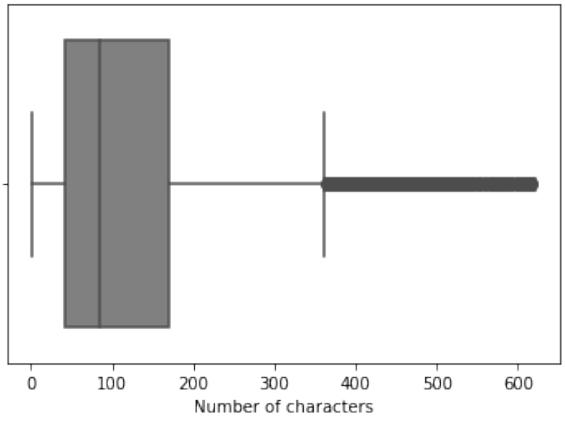

Figure 3: Boxplot of the distribution of the number of characters

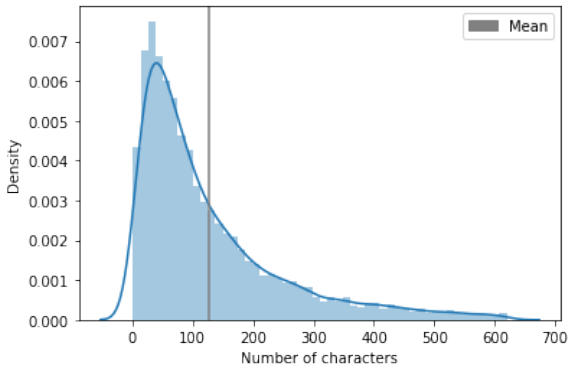

Figure 4: Probability distribution and histogram of the number of characters in the texts

4.3 Machine Learning Models

The classification models used to perform the binary classification are provided by the scikit-learn Python library (Pedregosa et al., 2011). The following models are used, as well as their fundamental hyperparameters:

- **Multinomial Naive Bayes**: alpha=1;
- **Logistic Regression**: Penalty L1 and L2, liblinear solver;
- **Support Vector Machines**: C=1, Linear and RBF kernel, gamma (RBF kernel)= $1/(N_{features} * var(X_{train}))$ and 10000 iterations as limit;
- **Decision Tree**: Based on CART and gini criterion, the leaf nodes of the final tree will have met the purity criterion by gini or will have only one child;
- **Random Forest**: 100 trees in the forest;
- **AdaBoost**: Depth decision tree 1 as weak estimator, 50 estimators and learning rate=1;
- **Gradient boosting**: Log loss (same used in logistic regressions) as loss function, mean square error with Friedman score as model performance criterion and learning rate 0.1;
- **Stacking classifier**: Predictions from SVM with linear kernel, logistic regression with L1 penalization and Random Forest as input, logistic regression with L2 penalization as final estimator.

To evaluate the classification models, the F1-score macro was used to identify the models with the best performance, but the AUC and the accuracy metrics were also calculated. F1 was chosen since this binary problem must weigh both the precision in correctly classifying an author, and recall, finding the model that combines sensitivity and specificity. Still about the evaluation of the models, pair-to-pair combinations of the set of 15 authors were made, to generate the 105 combinations of classification models trained with 75% of the oldest texts of each author, and tested with the 25% most recent. The resulting evaluation metrics for each model correspond to the average of 105 runs.

5 Authorship Attribution Models

Fig. 5 summarizes all the possibilities tested to evaluate an author attribution method through experiments with all possible binary combinations of authors. First, the training data are collected from a set restricted to two authors. With this, each comment goes to a vectorization approach that can be through TF-IDF, frequency, embeddings by word2vec or BERTimbau. If an embedding approach is used, each dimension of the word embeddings is averaged to generate a single text embedding. Also, when it comes to frequency weighting or TF-IDF, it is possible to replace the text terms with their respective parts of speech (since tokens here are the grammatical classes, only n-grams of words will be used in these cases).

For each classifier the metrics were evaluated in 16 vectorization options in each of 105 binary combinations of authors, resulting in 1680 combinations of authors and vectorizations in each of the 10 models and, consequently, 16800 different experiments. Word embeddings were scaled using the MinMaxScaler method, while those based on n-grams were scaled using the MaxAbsScaler method given the sparseness of the data. The scale did not change the results that much, since the scale for each vectorization does not differ that much.

6 Results and Discussion

Starting the analysis through the results of vectorization by TF-IDF and counting (Table 1), we can see that accuracy and F1-score macro have very close values, with a difference of less than 0.01 in all metrics. This pattern is repeated in the other vectorization methods as well, a consequence of both the high number of tests performed to generate the average of these values (105 experiments) and the fact that the data were delivered in a practically balanced way - approximately 750 textual data from each author to training and 250 test.

A value to ensure that the assignments between the two authors are clearly distinguished is the AUC-ROC (Area Under Curve - Receiver Operator Characteristic) metric. As the limit of the AUC metric is 1, values above 0.9 indicate that there is a clear distinction being made between the authors' texts.

The different combinations of textual representation by counting and TF-IDF demonstrated the need to test

Table 1: Average of metrics for feature extraction using frequency and TF-IDF.

Classifier	Best Text Representation	AUC	Accuracy	F1-score
Logistic Regression (L1)	TF-IDF for 1,2,3,4,5-grams of characters	0.9296	0.8617	0.8616
Logistic Regression (L2)	TF-IDF for words	0.9410	0.8723	0.8721
Multinomial Naive Bayes	1,2,3-grams counting	0.9216	0.8202	0.8166
SVM (linear kernel)	TF-IDF for words	0.9361	0.8671	0.8669
SVM (rbf kernel)	TF-IDF for words	0.8788	0.7765	0.7697
Random Forest	Words counting	0.9189	0.8510	0.8506
Decision Tree	1,2,3-grams counting	0.8274	0.8271	0.8268
AdaBoost	1,2,3,4,5-grams of characters counting	0.9221	0.8519	0.8517
Gradient boosting	1,2,3-grams counting	0.8274	0.8271	0.8268
Stacking	TF-IDF for words	0.9479	0.8825	0.8823

different vectorization methods, since half had better performance with counting/TF-IDF. The use of characters in turn was not as promising as the use of words, but one good result was with logistic regression with L1 penalty, where the selection of features carried out by the penalty allowed capturing only the sequences of relevant characters to the distinction between the authors. A case of range of word counting that worked well for this problem is the case of Naive Bayes which, due to its multinomial implementation, can generate a better decision rule for word frequency distributions in this problem.

The best performing method, based on the average of the F1-score macro, was the implementation of Stacking of classifiers. Since the methods that make up Stacking had good results independently, it was expected that together they would result in a minimally equivalent model, but, in fact, the combination of these different approaches achieved better results.

Table 2 shows the results referring to the text processing considering POS tagging using only unique/range of words. One of the first things to notice in these results is the significant change in the average F1-score for the classification, between 0.71 and 0.78 for all classifiers. Although it seems to be an inferior result, as it is, it must be considered that only the tokens related to the grammatical tags were treated, and consequently, only information from the textual structure can be extracted. Reaching results above 0.7 for F1-score and in some cases above 0.8 for AUC is an indication that even though all authors follow the same norm for speaking the language, there are style factors in composing the text structure that distinguished to a relevant degree. Thus, POS tagging is a a promising approach to work together with other text representation strategies.

With the exception of the RBF kernel SVM, the use of word ranges was the best textual representations for the tested models, which is a way found by the algorithms to perform the classification based on a broader field, i.e., on the relationship between the tokens that represent the parts of speech and how they appear to each author. For the case of the RBF kernel SVM, it is expected that the increase in token combinations has not contributed to being able to separate the data in a Gaussian way in the multidimensional space.

The best result based on the average was using the Gradient Boosting classifier, although very close to other methods. Linear separation still shows promise results, however, boosting methods are able to make a continuous improvement in weaker models that sometimes allow superior results.

Table 3 presents the results for the vectorization of

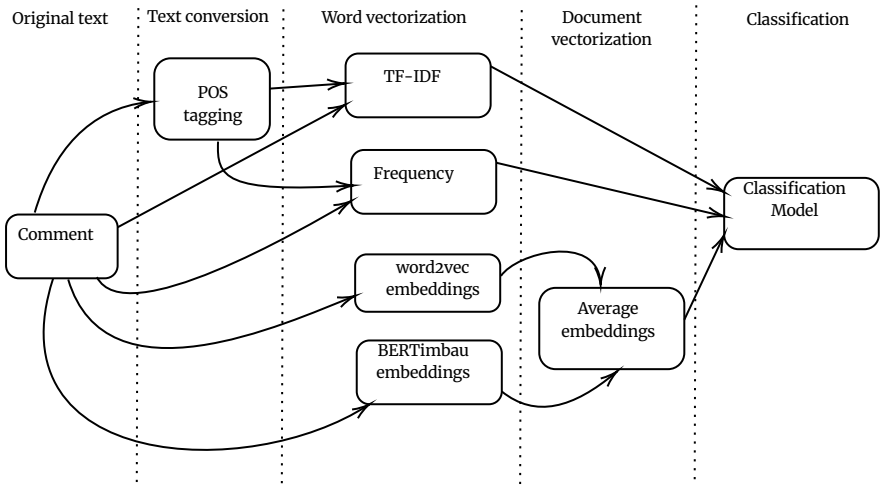

Figure 5: Experiments structure.

Table 2: Average of metrics for feature extraction using frequency and TF-IDF with POS tagging.

Classifier	Best Text Representation	AUC	Accuracy	F1-score
Multinomial Naive Bayes	TF-IDF for 1,2,3-grams of words	0.8002	0.7160	0.7131
Logistic Regression (L1)	TF-IDF for 1,2,3-grams of words	0.8444	0.7696	0.7693
Logistic Regression (L2)	TF-IDF for 1,2,3-grams of words	0.8366	0.7595	0.7591
SVM (linear kernel)	TF-IDF for 1,2,3-grams of words	0.7962	0.7314	0.7309
SVM (rbf kernel)	TF-IDF for words	0.8113	0.7432	0.7424
Decision Tree	1,2,3-grams counting	0.7141	0.7112	0.7108
Random Forest	TF-IDF for 1,2,3-grams of words	0.8385	0.7610	0.7599
AdaBoost	1,2,3-grams counting	0.8307	0.7588	0.7583
Gradient boosting	1,2,3-grams counting	0.8498	0.7729	0.7723
Stacking	TF-IDF for 1,2,3-grams of words	0.8399	0.7670	0.7667

Table 3: Average of metrics for feature extraction using word2vec

Classifier	AUC	Accuracy	F1-score
Multinomial Naive Bayes	0.7109	0.6509	0.6305
Logistic Regression (L1)	0.8256	0.7527	0.7507
Logistic Regression (L2)	0.8226	0.7480	0.7455
SVM (linear kernel)	0.8433	0.7698	0.7682
SVM (rbf kernel)	0.8363	0.7458	0.7416
Decision Tree	0.6963	0.6965	0.6961
Random Forest	0.8488	0.7742	0.7738
AdaBoost	0.8261	0.7556	0.7552
Gradient boosting	0.8490	0.7732	0.7729
Stacking	0.8425	0.7718	0.7707

Table 4: Average of metrics for feature extraction BERTimbau embeddings

Classifier	AUC	Accuracy	F1-score
Multinomial Naive Bayes	0.8342	0.7146	0.7068
Logistic Regression (L1)	0.9332	0.8711	0.8707
Logistic Regression (L2)	0.9392	0.8769	0.8766
SVM (linear kernel)	0.9253	0.8624	0.8621
SVM (rbf kernel)	0.9486	0.8853	0.8848
Decision Tree	0.7114	0.7117	0.7110
Random Forest	0.9058	0.8293	0.8283
AdaBoost	0.8888	0.8147	0.8143
Gradient boosting	0.9178	0.8453	0.8448
Stacking	0.9342	0.8732	0.8729

documents based on the average of the word embeddings from word2vec. The results are similar to the ones found in Table 2, with the exception of Naive Bayes Multinomial (which was not able to deal well with the data because it did not followed a multinomial distribution) and the decision tree, which acting as a strong estimator alone did not bring good results. The remaining classifiers ranged between 0.74 and 0.78.

Gradient boosting and Stacking classifier achieved equivalent results to Random Forest, although the forest resulted in a higher average F1-score. The performance of word2vec is also related to the basis on which the word embeddings were trained. The corpus of approximately 15,000 data examples was able to abstract enough context to keep the classifiers clearly performing better than the random classifier, but more data related to the social network would improve the embeddings formulation.

Boosting and stacking algorithms worked well for this problem, indicating that with a larger corpus composition it would be possible to further increase their performance. Random Forest worked in an equivalent way, it is worth mentioning it creates different trees acting as weak estimators to generate a significant performance classifier, and given that there is a smaller set of attributes to build the tree, it also influenced that the information collected by the trees were significant. With a relatively high F1-score, a classifier using vectorization by embeddings, i.e., dealing more with the context, is distinguished from the previous ones.

The results of the classifiers using the embeddings by BERTimbau (Table 4) showed a higher dispersion when compared to the other methods. The author attribution model based on the use of pre-trained embeddings from the BERTimbau architecture stands out in the context they abstract.

Logistic regressions are models that brought high results with all vectorization methods presented, as well as in BERTimbau. In case of evaluating the performance of text vectorization proposals, these regressions are good choices. With BERTimbau, the best performing classification model based on the F1-score was an SVM with RBF kernel. In this case, the linear classification of an SVM in the higher dimensional space was not enough, although very good, but the RBF kernel was able to distinguish with higher F1-score one author from another based on a Gaussian separation in a simulated high-dimensional space by RBF kernel.

Fig. 7 summarizes the macro F1-score data extracted from the previous tables with the addition of the confidence interval. The models based on n-grams of characters and words that used the source text maintain a higher level of F1 score, but this implies that all models with higher F1 are contained in the same confidence intervals, therefore, being equivalents. This pattern of many classifiers with the same significance level follows with word2vec and vectorization in texts converted to POS tagging, but with lower F1-score values. Embeddings by BERTimbau differ a little because the confidence interval and the average performance of the classifier vary more, yet it is possible to notice the RBF kernel SVM with BERTimbau stands out, which, however, enters the same level of significance as Stacking and L2 logistic regression.

To summarize the analysis, we collected the model of each vector that achieved the best result in its significance

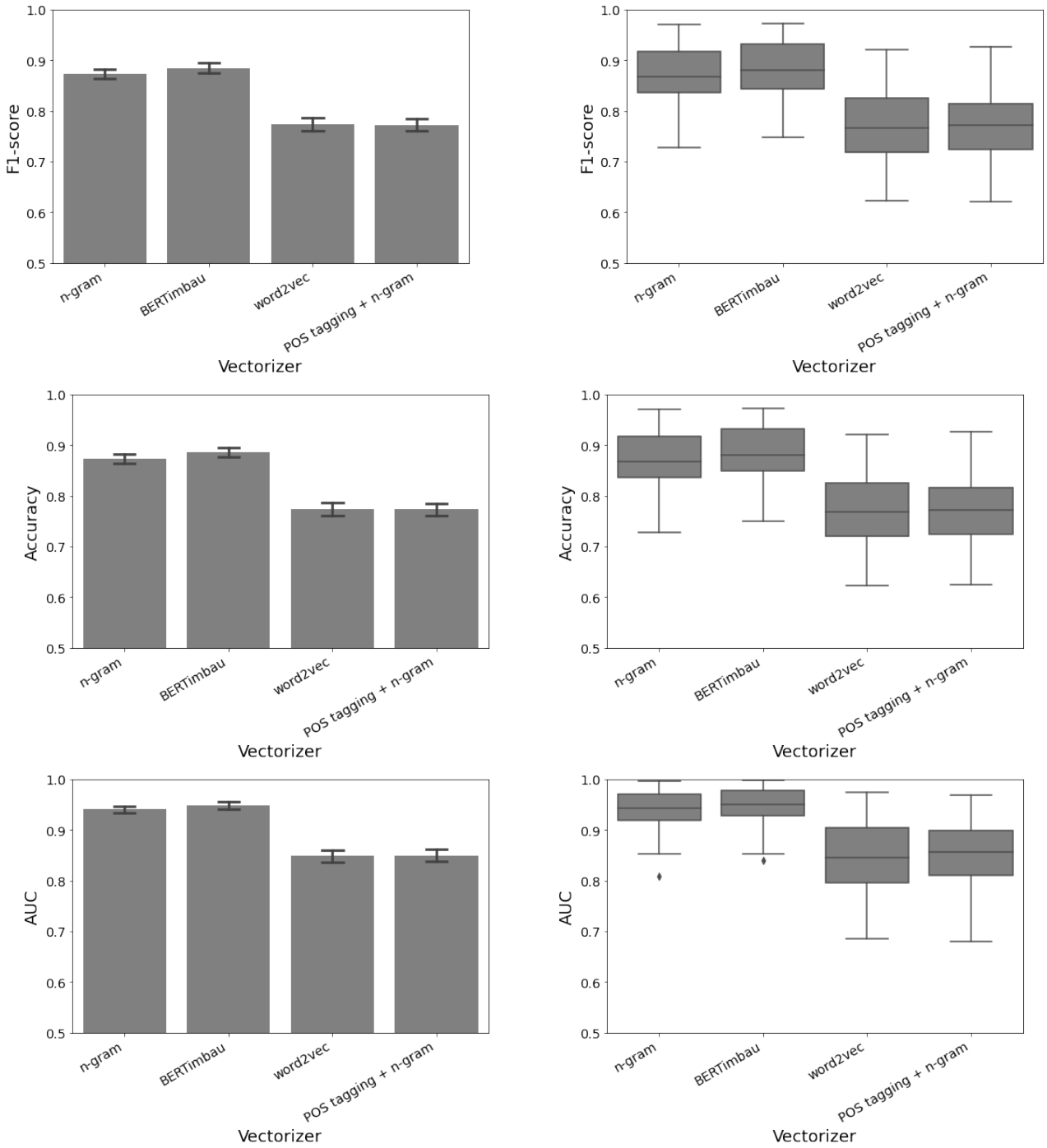

Figure 6: Best models according to the average of the macro F1-score metric for each vectorizer. For the macro F1-score, accuracy and AUC metrics, the mean and 95% confidence interval are shown in the bar graphs on the left, while the graphs on the right show the dispersion of these same data using a boxplot.

level, which are the ones with the highest average F1-score. Fig. 6 shows the confidence interval and the dispersion of the best models for vectorization by n-grams, POS tagging with n-grams, word2vec and BERTimbau, being, respectively, Stacking Classifier, Gradient boosting, Random Forest and RBF kernel SVM.

There is a clear equivalence between authorship attribution with n-grams and BERTimbau, as well as in vectorization by word2vec and POS tagging with n-grams. The dispersion of the two models with the highest F1-score are very similar and, when it comes to AUC, they approach 1 in some cases. The use of classical techniques,

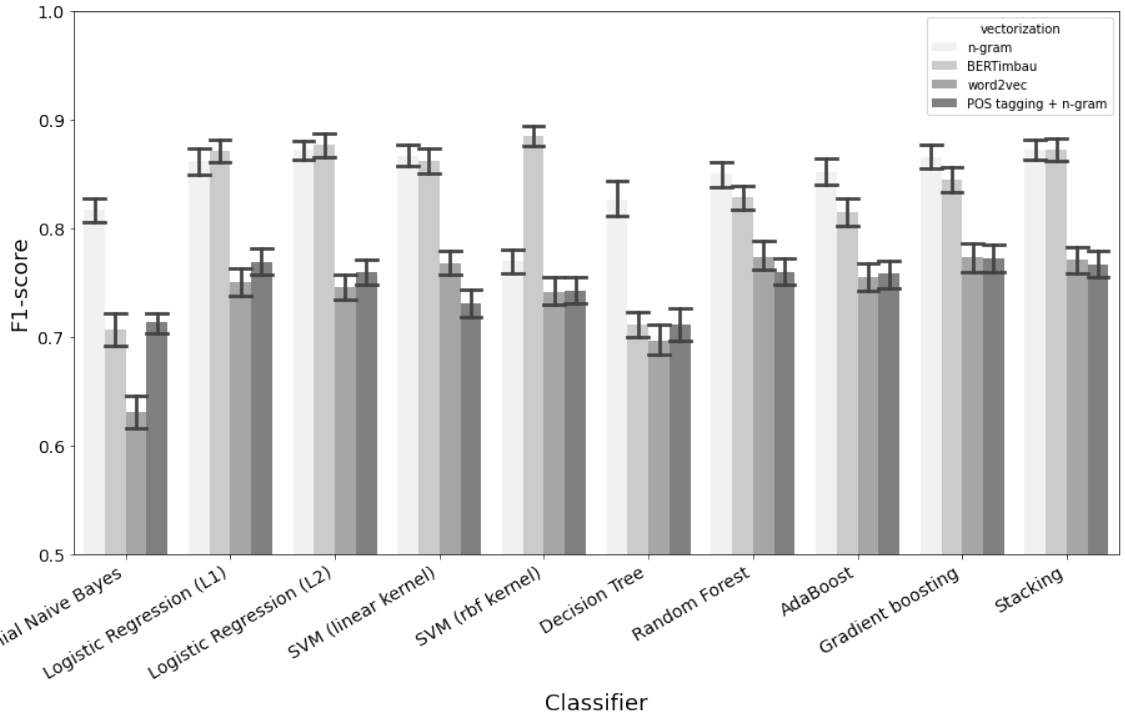

Figure 7: Average of the macro F1-score metric for the runs for the 10 classifiers and the four vectorization methods. In the case of extracting attributes by n-grams using the original text or the conversion by post tagging, only the method that resulted in the highest average F1-score macro is displayed. The error bars accompanying the mean of each classifier represent the lower and upper limits of the 95% confidence interval for each experiment.

precisely because they are simpler to implement and execute, still manages to bring results equivalent to those of more sophisticated methods. To improve the models, it is possible to use methods that observe other aspects of the text and thus detect author styles, as seen with the case of POS tagging that, even without metrics comparable to the two higher ones, seem to bring value when combined with other methods.

7 Conclusion and Future Works

Authorship attribution is a task that has been analyzed for years and that has high importance in social network environments. Considering the methods assessed, divergences and equivalences were noted. POS tagging is an important tool to be studied together with other methods that analyze the context and thus allow a more complete analysis to perform authorship analysis, as identified in this paper. Still, sophisticated methods such as those based on transformers bring satisfactory results for the task, however, equivalent to classic ways of extracting textual attributes by counting or TF-IDF. Studies with larger corpus in Portuguese in order to use the word2vec approach tend to bring better representations of context and consequently improve the performance of methods based on embeddings trained in the corpus.

The present work showed the importance of text processing for the task of binary authorship attribution compared to the choice of the model, which had importance, but has less influence for the determination of significant results. The distinction of authors identified through an AUC 0.94 and an F1-score of 0.88 demonstrates that there is already a high level of confidence to authorship attribution models in social networks such as Reddit with simple and complex methods, but that can be improved with new approaches for extracting attributes from text.

This work stands out concerning its counterparts in Portuguese, as it presents a comparative analysis of several current techniques, including the use of LLM, and statistically evaluates the difference between the results obtained.

Present work is limited to the scope of its application, that is, data in Portuguese from the social network Reddit, tested considering 15 authors who had posted at least 1,000 posts. Although the results consider posts without a character limit, it was observed that most of them had less than 128 characters. Thus, it is likely that the results obtained here are also valid for social networks or microblogs in which posts are limited to a reduced number of characters.

References

Abbasi, A. and Chen, H. (2005). Applying authorship analysis to extremist-group web forum messages, *IEEE Intelligent Systems* **20**(5): 67–75. https://doi.org/10.1109/MIS.2005.81.

Boe, B. (2022). PRAW: The Python Reddit API Wrapper, Github repository. Available at https://github.com/praw-dev/praw.

Casimiro, G. R. and Digiampietri, L. A. (2020). Authorship attribution using data from reddit forum, *XVI Brazilian Symposium on Information Systems*, SBSI'20, Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411564.3411616.

Casimiro, G. R. and Digiampietri, L. A. (2022). Authorship attribution with temporal data in reddit, *Proceedings of the XVIII Brazilian Symposium on Information Systems*, SBSI '22, Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3535511.3535515.

Custódio, J. E. and Paraboni, I. (2019). An ensemble approach to cross-domain authorship attribution, *in* F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D. E. Losada, G. Heinatz Bürki, L. Cappellato and N. Ferro (eds), *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Springer International Publishing, Cham, pp. 201–212. https://doi.org/10.1007/978-3-030-28577-7_17.

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)*, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423.

Honnibal, M., Montani, I., Van Landeghem, S. and Boyd, A. (2022). spacy: Industrial-strength natural language processing in python. http://doi.org/10.5281/zenodo.1212303.

Layton, R., Watters, P. and Dazeley, R. (2010). Authorship attribution for twitter in 140 characters or less, *2010 Second Cybercrime and Trustworthy Computing Workshop*, pp. 1–8. https://doi.org/10.1109/CTC.2010.17.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space, *Proceedings of Workshop at ICLR* **2013**. https://doi.org/10.48550/arXiv.1301.3781.

Nothman, J., Ringland, N., Radford, W., Murphy, T. and Curran, J. R. (2013). Learning multilingual named entity recognition from wikipedia, *Artificial Intelligence* **194**: 151–175. https://doi.org/10.1016/j.artint.2012.03.006.

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. Available at https://dl.acm.org/doi/10.5555/1953048.2078195.

Rademaker, A., Chalub, F., Real, L., Freitas, C., Bick, E. and de Paiva, V. (2017). Universal Dependencies for Portuguese, *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, Linköping University Electronic Press, Pisa,Italy, pp. 197–206. Available at https://aclanthology.org/W17-6523.

Souza, F., Nogueira, R. and Lotufo, R. (2020). Bertimbau: Pretrained bert models for brazilian portuguese, *in* R. Cerri and R. C. Prati (eds), *Intelligent Systems*, Springer International Publishing, Cham, pp. 403–417. https://doi.org/10.1007/978-3-030-61377-8_28.

Swain, S., Mishra, G. and Sindhu, C. (2017). Recent approaches on authorship attribution techniques — an overview, *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Vol. 1, pp. 557–566. https://doi.org/10.1109/ICECA.2017.8203599.

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* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett (eds), *Advances in Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc. https://doi.org/10.48550/arXiv.1706.03762.