# Brazilian Fake news detection using TrollBR

Isabelle Rodrigues Vaz de Melo
*Programa de Engenharia Elétrica*
*Universidade Federal do Rio de Janeiro*
Rio de Janeiro, Brasil
isabelle.melo@coppe.ufrj.br

***Abstract*—With the rise of the internet over the years and globalization, it has become easier to communicate in a more dynamic way. Millions of pieces of information circulate daily on the internet and reach all possible audiences with the most diverse opinions. Although this facilitates personal, work, social and economic relationships, it can have negative impacts on the freedom people have to share content on the internet. Often, social networks and websites do not have a filter of what can or cannot be published, and this ends up giving rise to fake news being published.**

**Fake news spread quickly and can impact the economic and social scenario of a nation, influencing masses and distorting reality.**

**The current work explores the use of four Natural Language processing methods to classify Brazilian Fake news from a new dataset: TrollBR. Logistic regression applied to embeddings of a TF-IDF vectorizer [4], BERTimbau-base [5] and BERT-base [6] in English are used and evaluated, and Artificial neural networks [8]. In addition, the winning results are compared with results from previous works on classification of Brazilian news and a discussion is made on the quality of the methods chosen as possible candidates for an automatic detection tool for fake news.**

***Index Terms*—NLP, Fake News, BERT, Deep learning**

## I. Introduction

In 2016 - the year of the presidential elections in the United States of America - the term fake News began to become popular in the face of a scenario in which the internet was being used as the main means of bombing false news. In several social networks, mainly on Twitter, false news with appealing natures were reported daily, encouraging a certain bias.

At the time, most of them were of a political nature, appearing to manipulate and coerce the American electoral process. Despite this, fake News has been spread since the beginning of the media with the ultimate goal of distorting reality for the benefit of those who produce it. With the rise of the internet and its accessibility around the world, anonymity has become a powerful weapon for spreading hate speech and fake news, generating various socio-economic impacts.

As a result, millions of news items are generated every day and spread in chains through user shares on the internet, as it's shown in Figure 1. It is practically impossible for each one of them to be checked individually, so it is necessary to investigate other methods for its detection. The purpose of this research is to investigate the effectiveness of Natural Language Processing (NLP) models in a new dataset called TrollBR for classifying if the news are real or fake. The dataset is a combination of three other Brazilian fake news datasets: FakeRecogna [1], Fake.Br [2] and FACTCKBR [3]. The investigated methods are: the use of an embedding generated by a TF-IDF vectorizer [4] classified by a Logistic regression as baseline, Artificial neural networks are also tested for comparsion effects, and classification by two pre-trained models that use Transformers - BERTimbau-base [5] and BERT-base english [6].

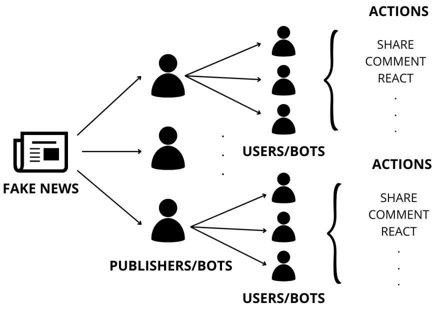

Fig. 1. Fake news spreading Schematic.

The final results are compared by accuracy with some previous works that are cited in the next section. There are several more well-founded works in the English language that use transformer models, such as [18][19][20], however the studies in Portuguese language have not yet been very well explored and there is also not a large amount of public data available.

## II. Previous works

In this section, a brief discussion is made of results found in similar previous works that made use of one or more of the datasets used in this work.

- In the work ”Contributions to the Study of Fake News in English: New Corpus and Automatic Detection Results” [2] the Corpus Fake.BR is presented. With this Corpus, several combinations of embeddings and different features are generated, applying a linear SVC with default parameters for news classification and truncating the longer texts to the size of their aligned counterparts.
  Using Bag of Words features [8] led to the best accuracy result by SVC [10]: 88%. Changing SVC by a Multiple Layer perceptron increased the results in 2%.

| Dataset | Accuracy |
|---|---|
| Fake.Br | 95.4% |
| Fake.Br + FakeRecogna | 94.6% |
| FACTCK.BR | 87.7% |
| FACTK.BR + FakeRecogna | 93.3% |

TABLE I
ACCURACY SCORES FROM [1]

Running the experiments without truncating the size of the texts achieved 96% of accuracy with Bag of words, but this classification was probably biased, as true texts were significantly longer than the fake ones and the dataset is not fully balanced.

- The work [1] show some results tested in FakeRecogna and the mix of FakeRecogna and Fact.Br with preprocessing. Using FakeRecogna Corpus, generating embeddings from Bag of words [8] and FastText [10], the best Bow accuracy result was 93.1%. The best FastText accuracy found was also from MLP: 84.8%. A Convolutional neural network was tested and it achieved a 94.8% accuracy. Table I shows accuracy results using a Convolutional neural network found.

## III. Models investigated

- TF-IDF + Logit: The first investigation attempt consists in defining a baseline. Given that the problem is a supervised learning one, a technique for generating textual embeddings that is not associated with transformers is used and a classification of these embeddings is performed with a Logistic regression. According to [4], the frequency–inverse document frequency method, also called TF-IDF, is a statistical measure that aims to quantify the importance of a word in a document in relation to a collection of documents or in a linguistic corpus.
Term frequency works by analyzing the frequency of a particular term relative to the document. Inverse document frequency makes possible to determine how common or uncommon a word is amongst the entire corpus, such that $t$ is the word to measure it's commonness and $N$ is the number of documents ($d$) in the corpus ($D$). The denominator in Equation 1 indicates the number of documents the term $t$ appears in:

$$idf(t, D) = log(\frac{N}{count(d \in D : t \in d)}) \tag{1}$$

Taking inverse document frequency can minimize the weighting of frequent terms while making infrequent terms have a higher impact. TF then shows the information of how often a term appears in a document, and IDF the information of rare words counting. Multiplying both terms, it's possible to obtain TF-IDF mathematical form (Equation 2):

$$tfidf(t, d, D) = tf(t, D)idf(t, D) \tag{2}$$

The higher the TF-IDF score the more relevant the term is; as a term gets less relevant, its TF-IDF score will approach 0.
This procedure generates embeddings, that is, vector representations of the words in the text. For each text together, we have a respective embedding. A classification is then made via logistic regression using these dense vectors as input - as shown in Figure 2.

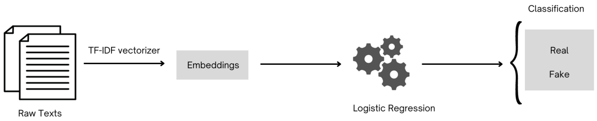

Fig. 2. TF-IDF + Logit Schematic.

- Artificial neural networks: Effective and mostly used methods in the field of Natural language processing are those that use Deep Learning models.
Deep learning models have the ability to automatically extract features, maximizing model performance for the task at hand. Its use is made through artificial neural networks with several layers called hidden layers. Figure 3 shows a scheme of the mechanism that occurs from an entry into a neural network node.

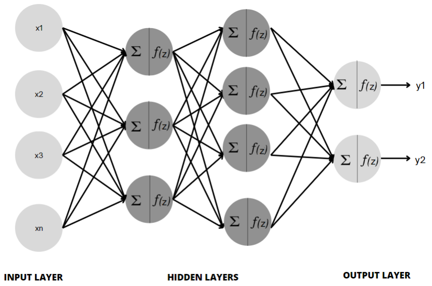

Fig. 3. Simple Neural network with 2 layers Schematic.

The inputs are linearly combined with weights and a bias term, and then passed through a non-linear activation function (such as a ReLu, Hyperbolic Tangent or sigmoid function). This allows a separation of the data in order to perform the classification such that:

$$z = \sum_{i=1}^{n} w_i x_i \tag{3}$$

$$f(z) = f(\sum_{i=1}^{n} w_i x_i) \tag{4}$$

and f (z) represents the application of an activation function. This mechanism goes trough the layers, reaching a final output point and then stepping the backward pass so weights can be updated based in the final loss.
The neural network approach used takes into account the sequence of words. Text are a kind of time series. Phrases are a sequence of words that form meaning, and word order is crucial in forming meaning. Words can also change meaning depending on the context. Sequential

approach allows for better context capture. One of the layers, the Embedding layer, is responsible for mapping the words that have semantic proximity, having their own weights, as well as common neurons, and can be trained along with the rest of the neural network.

It's also applied a Max Pooling to reduce the dimensionality of the Embedding layer's output vector. This drastically reduces the computational cost of neural network training. Max Pooling can be understood as a layer that summarizes information. Finally, Dropout is applied.

Dropout is a layer that during training randomly disables some synapses that exist between one layer of the network and the other. When a synapse is disabled, its weight is also not changed by the optimizer. The purpose of Dropout is overfitting prevention.

- Bidirectional Encoder Representations from Transformers (BERT): BERT is a model released by Google AI Language [6] in 2018. It makes use of Transformer, an Attention mechanism that learns contextual relations between sub-words in a text.

  Attention mechanism works in such a way to correct Recurrent Neural Networks limitations [8]: Due to RNN's architecture, it's impossible to encode information from all the input timesteps because they tend to forget information from timesteps that are far behind, making it hard to generate a contextual vector that captures the most important information from input sentences.

  In the encoder-decoder, it's possible to define a previous state as $y_{i-1}$ and hidden states as $h_1, h_2, ..., h_n$, such that:

  $$e_i = attention(y_{i-1}, \mathbf{h}) \in \mathbb{R}^n \tag{5}$$

  Where $i$ index indicates the prediction step, it is, defines a score between the decoder hidden state and encoder hidden states. For each hidden state, denoted by $j$ index, a scalar is calculated to perform attention weights:

  $$e_{ij} = attention(y_i, h_j) \tag{6}$$

  But since the outputs must be probabilities distributions, a Softmax is applied in $e_{ij}$ to get real attention weights:

  $$\alpha_{ij} = \frac{exp(e_{ij})}{\sum exp(e_{ik})} \tag{7}$$

  Attention, then, is defined as the weighted average of values, and this weighting is a learned function. $\alpha_{ij}$ is a data-dependent dynamic weight. Figure 3 shows an Attention schematic.

  In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. The BERT architecture is composed of several Transformer encoders stacked together. Further, each Transformer encoder is composed of two sub-layers: a feed-forward layer and a self-attention layer. The Transformer encoder reads the entire sentence at once as it is bidirectional, allowing the model to learn all surroundings of words to better understand the context.

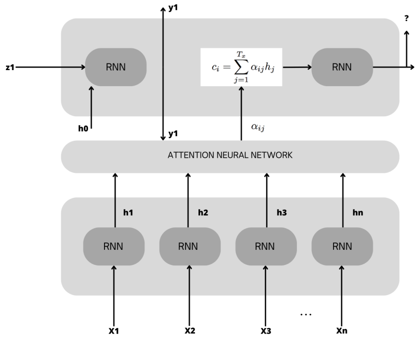

Fig. 4. Attention mechanism Schematic.

  It then generate contextualized embeddings, and they pass through a classifier (usually a feed-forward neural network + softmax, although it's architecture can be changed in multiple forms) generating output probabilities of a task.

  It is used in Sentiment analysis, Question answering, Text prediction, Text generation, Summarization, and Polyssemy resolution. The model works by recieving a large amout of data (3.3 Billion words, trained on Wikipedia - 2.5B words - and Google's BooksCorpus - 800M words). The BERT models used in this work were pre-trained, and then a fine-tuning was performed for the Sentence classification task - between fake or real.

  BERT-base [6] in English, and BERTimbau [5] were used. BERT-base has 110 million parameters, BERTimbau-base also has 110 million parameters and 12 layers of Transformers blocks with a hidden size of 768 and number of self-attention heads, in both cases. The maximum sentence length is set to 512 tokens in both of them. To assess the coherence of the results of both models with Transformers, BERT embeddings are also generated, and these pass through a Logistic regression classifier.

## IV. Experiments

### A. Database

As mentioned in Section 1, TrollBR is a combination of 3 other datasets: FakeRecogna [1], Fake.Br [2] and FactBR [3]. Some important information about the datasets is that they were all collected by the respective authors and labeled manually. They contain a series of tabular information such as URL, author, date, text, label (real or fake), and category. For the work, only the features of text, category and label were used. There were nine categories: health, Brazil, politics, economy, science, entertainment, world and religion, all collected between 2018 and 2022. Table II shows the relation between number of texts, real and fake news for each database.

To use the BERT-base in English, it was necessary to translate all the texts into Portuguese. For this, the Google translate API was used and the new texts in English were generated. Given the character limitations of Google translate,

|  | FakeRecogna | FakeBR | FactBR | TrollBR |
|---|---|---|---|---|
| Texts | 11872 | 7200 | 1313 | 20385 |
| Fake news | 5936 | 3600 | 387 | 9785 |
| Real News | 5936 | 3600 | 926 | 10600 |

TABLE II
RELATION BETWEEN DATABASES

the sentences were reduced to 4000 characters and there were also some texts that were not supported by the API due to formatting issues - these were excluded. In the end, the English dataset was left with 20358 texts.

Figure 5 shows the category bar graph of topics mentioned in TrollBR. As it can be seen, politics is the main theme of most news and Figure 6 shows the word cloud associated with the most mentioned words. These words are highly associeted with politics and Brazil's presidential elections of 2018 and 2022 (stop words in portuguese) have been removed.

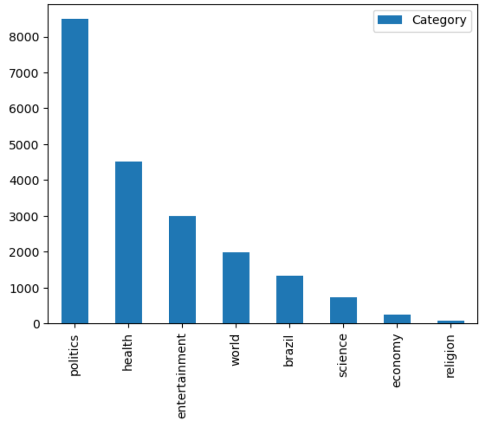

Fig. 5. Bar graph of categories in TrollBR (general).

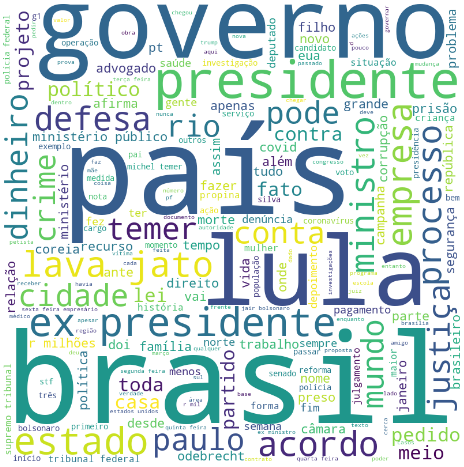

Fig. 6. Word cloud TrollBR.

Figure 7 shows the numbers of news per category being fake. Most of fake news are associated with politics and health. The author suspect this happens due the big amount of news spread during 2018 and 2022 elections in Brazil, and the COVID-19 pandemics.

| C | Penalty |
|---|---|
| [0.001, 0.01, 0.1, 1, 10, 100, 1000] | [l1, l2, elasticnet, none] |

TABLE III
HIPERPARAMETERS TF-IDF + LOGIT

Sentence sizes were also evaluated: in the general set, the longest sentence has 46084 tokens, while the shortest has 46.

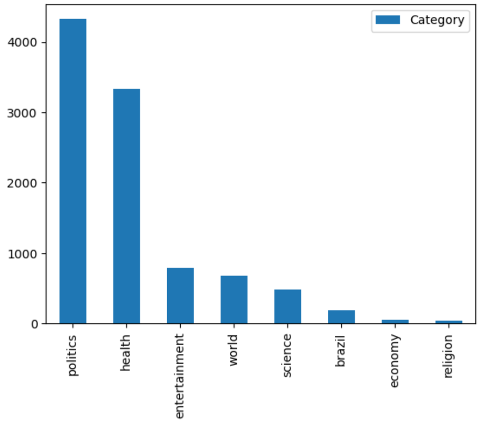

Fig. 7. Bar graph of categories in TrollBR (fake news).

### B. Pre-processing

- TF-IDF + Logit: Lowering texts, eplacing everything with space except (a-z, A-Z, ".", "?", "!", ","), removing URLs, html tags, punctuations and stopwords.
- Artificial neural networks: Lowering texts, eplacing everything with space except (a-z, A-Z, ".", "?", "!", ","), punctuations, stopwords, truncation of texts to equal length (size 512).
- BERTimbau-base and BERT-base: Lowering texts, eplacing everything with space except (a-z, A-Z, ".", "?", "!", ","), punctuations, stopwords, truncation of texts to equal length (size 512).

### C. Training and Validation

For training and validation, the data were divided in the proportion 80:10:10 (training, validation and test). For each model, the following hyperparameters were investigated:

- TF-IDF + Logit: Table III shows hiperparameters investigated.
- Artificial neural network: Table IV shows hyperparameters investigated for NN's. The NN used in this work has 2,480,247 parameters and 4 layers: The embedding one, Max pooling, dropout and a dense layer - just a regular densely-connected NN layer.
- BERTimbau-base and BERT-base: The hyperparameters that were used for fine-tuning followed the recommendation of the BERT article [6] to test batch sizes of 16 or 32, epochs between 2 and 4, and learning rates between

| Batch size | Epochs | Dropout |
|---|---|---|
| [32] | [11,60] | [0.3, 0.1, 0.5,0.05] |

TABLE IV
HYPERPARAMETERS NEURAL NETWORK

| Epochs | Batch size | Learning rate |
|---|---|---|
| [2,3,4] | [16] | [2e-5,5e-5] |

TABLE V
HIPERPARAMETERS BERTIMBAU AND BERT

2e-5 to 5e-5. Due to computational limitations (Google Colab Tesla T4 12GB), some of these were reduced, as it's shown in Table V.

### D. Results

- TF-IDF + Logit: In order to explore possible results, the classifiers were tested in the isolate data sets, possible pair combinations of the dataset, and the final combination that forms TrollBR.
  To compare these results, no Grid search cross-validation was performed. The Table VI shows the results found. It is noticed that the accuracy for FACTCK.BR is the lowest of all. It is assumed that this occurs because in this dataset there is a considerable imbalance of classes, so it is not a good strategy to evaluate it by accuracy.
  It can also be noted that TrollBR's result was not the highest among all, losing only by 2% from the test on FakeRecogna.
  The author assumes that this occurs because, as three different data sets were joined - with slightly different sizes and text formats - it is expected that there will be a small drop in the final accuracy percentage.
  However, with more data, there is the possibility that the model will not be restricted to learning a single Corpus pattern, but rather be able to more clearly identify what is and is not real based on a large amount of data.
  After this step, TrollBR was chosen for its greater data variability and good performance. A Grid Search with cross validation was then performed. Table VII shows the best hyperparamters found using TF-IDF and test accuracy.

| Data set used | acc |
|---|---|
| Fake.Br | 95.0% |
| FakeRecogna | 96.0% |
| FACKT.BR | 84.0% |
| Fake.Br + FakeRecogna | 94.0% |
| Fake.Br + FACKT.BR | 93.0% |
| FakeRecogna + FACKT.BR | 94.0% |
| TrollBR | 93.0% |

TABLE VI
RESULTS TF-IDF + LOGIT COMPARING ALL THE DATA AVALIABLE

| C | Penalty | acc |
|---|---|---|
| 10 | l2 | 94.0% |

TABLE VII
RESULTS TF-IDF + LOGIT

| Batch size | Epochs | Learning Rate | acc |
|---|---|---|---|
| 16 | 2 | 2e-5 | 98.3% |
| 16 | 2 | 5e-5 | 98.4% |
| 16 | 3 | 2e-5 | 98.1% |
| 16 | 3 | 5e-5 | 98.5% |
| 16 | 4 | 2e-5 | 98.2% |
| 16 | 4 | 5e-5 | 98.5% |

TABLE VIII
RESULTS BERTIMBAU-BASE

| Batch size | Epochs | Learning Rate | acc |
|---|---|---|---|
| 16 | 2 | 2e-5 | 96.8% |
| 16 | 2 | 5e-5 | 97.5% |
| 16 | 3 | 2e-5 | 97.4% |
| 16 | 3 | 5e-5 | 97.1% |
| 16 | 4 | 2e-5 | 96.4% |
| 16 | 4 | 5e-5 | 97.1% |

TABLE IX
RESULTS BERT-BASE

- BERTimbau base: Table VIII shows the hyperparameters investigated and the accuracy results.
- BERT-base (english): Table IX shows the hyperparameters investigated and the accuracy results.
- Embeddings: Table X shows the accuracy results obtained from passing BERT embeddings through a Logistic Regression classifier.

Figure 8 shows a graph of accuracy from best results found and Figure 9 shows the results comparing trained models (pretrained and finetuned) with not trained models.

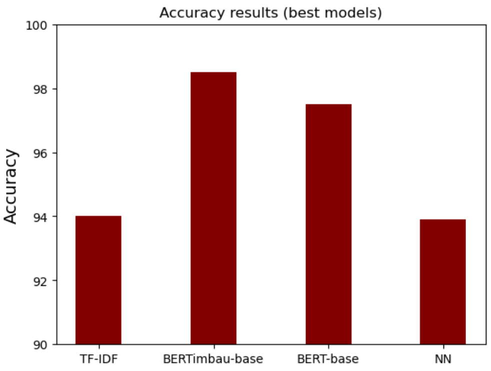

Fig. 8. Accuracy results (Best models).

- Artificial neural network: Table XI shows the results for NN's with the investigated hyperparameters.

### E. Comparing with previous works results

In this subsection, the best model found in the previous section - BERTimbau-base - is compared with the best models in the previous works section. Figure 10 shows the comparison.

| | BERTimbau embeddings | BERT embeddings |
|---|---|---|
| acc | 94.0% | 89.0% |

TABLE X
RESULTS BERT EMBEDDINGS + LOGIT

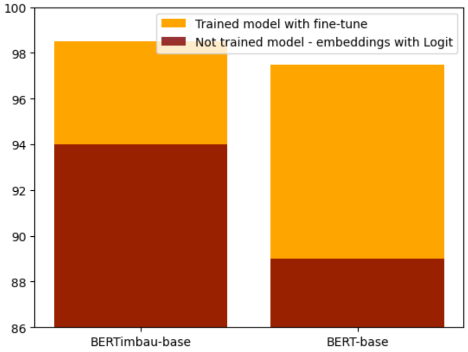

Fig. 9. Accuracy results (Best models vs Not trained models).

| Batch size | Epochs | Dropout | acc |
|---|---|---|---|
| 32 | 60 | 0.3 | 91.3% |
| 32 | 11 | 0.3 | 93.6% |
| 32 | 11 | 0.1 | 93.9% |
| 32 | 11 | 0.5 | 93.0% |
| 32 | 11 | 0.05 | 93.6% |

TABLE XI
RESULTS NEURAL NETWORKS

Table XII also shows the results. The comparison, however, has some minor nuances.

The first point is that in previous works, the three databases were not used together, and in the model currently used, TrollBR is used. The second point is that after this data increase, a Transformer model is tested against the machine learning models mentioned and a convolutional neural network. Despite this, it is still possible to state that TrollBR combined with BERT-type models provides better results than those previously found in the literature.

It is assumed that using more parameters in the model, such as using BERT-large, can even improve the results, taking it closer and closer to the maximum accuracy of 100%.

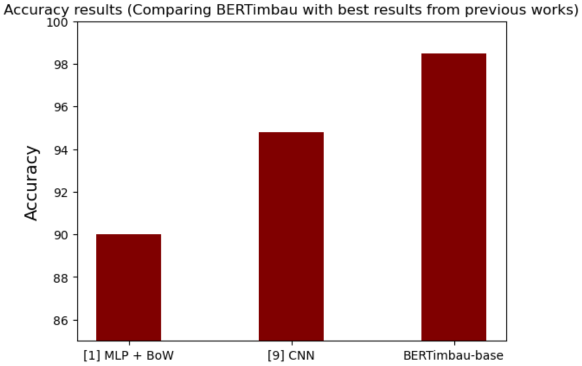

Fig. 10. Accuracy results (Best model vs Previous works).

## V. CONCLUSION

The study carried out brings up the possibility of implementing an automatic Fake news detector that comes close to 100% accuracy using BERT variations. When choosing the hyperparameters for BERTimbau, the model with batch size 16, 3 epochs and learning rate of 5e-5 was chosen because it has a lower computational cost compared to the model with batch size 16, 4 epochs and learning rate of 5e-5.

|  | [1] MLP + BoW | [9] CNN | BERTimbau-base |
|---|---|---|---|
| acc | 90.0% | 94.8% | 98.5% |

TABLE XII
ACCURACY SCORES - COMPARSION WITH PREVIOUS WORKS

The best result among all the tests was the BERTimbau-base with a final accuracy of 98.5%. This result (as well as the baseline BERT) is in tune with the baseline that used the TF-IDF vectorizer. BERTimbau-base increased the TF-IDF result by 4.5%, having been considered a great option to catch as much fake news as possible.

The assumption made is that using BERT variations and increasing the amount of data (TrollBR) in relation to previous works, the accuracy results would have a significant increase in the comparative aspect. In fact, better results were obtained than those reported in the literature for problems treated in Brazilian Portuguese.

Possible projections for the continuation of the current work are:

- Use k-fold techique to guarantee stability of results
- Test 32-batch size combinations
- Test more BERT variations, such as RoBERTa and ALBERT
- Analyze how much of each category corresponds to true or false news after the models' predictions
- Test BERT-large and BERTimbau-large
- Test BERT's accuracy for the possible combinations between the three data sets, and not just TrollBR
- Use LSTM architectures and explore it's hyperparamters to generate better context vectors
- Create an online platform for users to check fake news in real time

It is also possible to conclude that it is not worth using Artificial neural networks the way they were presented. They perform slightly worse results than the model with TF-IDF and are more computationally expensive.

For practical applications, the best options so far are to use TF-IDF to generate embeddings and apply a classifier (other classifiers can be tested) precisely because it is not so computationally expensive to train and get good results, or to use BERTimbau with the hyperparameters chosen for having a very high accuracy result, although it is more costly for training.

## REFERENCES

[1] Garcia, G.L., Afonso, L.C.S., Papa, J.P. (2022). FakeRecogna: A New Brazilian Corpus for Fake News Detection. In: , et al. Computational Processing of the Portuguese Language. PROPOR 2022. Lecture Notes in Computer Science, vol 13208. Springer, Cham.

[2] MONTEIRO, R. A. et al. Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In: SPRINGER. International Conference on Computational Processing of the Portuguese Language. [S.l.], 2018. p. 324–334.

[3] João Moreno and Graça Bressan. 2019. FACTCK.BR: a new dataset to study fake news. In Proceedings of the 25th Brazillian Symposium on Multimedia and the Web (WebMedia '19). Association for Computing Machinery, New York, NY, USA, 525–527.

[4] (2011). TF–IDF. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.

[5] Souza, F., Nogueira, R., Lotufo, R. (2020). BERTimbau: Pretrained BERT Models for Brazilian Portuguese. In: Cerri, R., Prati, R.C. (eds) Intelligent Systems. BRACIS 2020. Lecture Notes in Computer Science(), vol 12319. Springer, Cham.

[6] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[7] Sherstinsky. (2020). Elsevier BV. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena.

[8] Qader, Wisam M. Ameen, Musa Ahmed, Bilal. (2019). An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges. 200-204. 10.1109/IEC47844.2019.8950616.

[9] Fischer, M., Haque, R., Stynes, P., Pathak, P. (2022). Identifying Fake News in Brazilian Portuguese. In: Rosso, P., Basile, V., Martínez, R., Métais, E., Meziane, F. (eds) Natural Language Processing and Information Systems. NLDB 2022. Lecture Notes in Computer Science, vol 13286. Springer, Cham.

[10] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of Tricks for Efficient Text Classification

[11] Hochreiter, Sepp Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[12] Endo PT, Santos GL, Silva I, Egli A, Lynn T. COVID-19 Fake News in Brazilian Portuguese Language. Encyclopedia. Available at: https://encyclopedia.pub/entry/22439. Accessed December 13, 2022.

[13] M. Paixão, R. Lima and B. Espinasse, "Fake News Classification and Topic Modeling in Brazilian Portuguese," 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2020, pp. 427-432.

[14] Endo PT, Santos GL, de Lima Xavier ME, Nascimento Campos GR, de Lima LC, Silva I, Egli A, Lynn T. Illusion of Truth: Analysing and Classifying COVID-19 Fake News in Brazilian Portuguese Language. Big Data and Cognitive Computing. 2022; 6(2):36.

[15] Marcos Paulo Moraes, Jonice de Oliveira Sampaio, and Anderson Cordeiro Charles. 2019. Data mining applied in fake news classification through textual patterns. In Proceedings of the 25th Brazillian Symposium on Multimedia and the Web (WebMedia '19). Association for Computing Machinery, New York, NY, USA, 321–324.

[16] Galhardi CP, Freire NP, Minayo MCS, Fagundes MCM. Fact or Fake? An analysis of disinformation regarding the Covid-19 pandemic in Brazil. Cien Saude Colet. 2020 Oct;25(suppl 2):4201-4210. Portuguese, English.

[17] Szczepański, M., Pawlicki, M., Kozik, R. et al. New explainability method for BERT-based model in fake news detection. Sci Rep 11, 23705 (2021).

[18] Kaliyar, R.K., Goswami, A. Narang, P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. Multimed Tools Appl 80, 11765–11788 (2021).

[19] Nishant Rai, Deepika Kumar, Naman Kaushik, Chandan Raj, Ahad Ali, Fake News Classification using transformer based enhanced LSTM and BERT, International Journal of Cognitive Computing in Engineering, Volume 3, 2022, Pages 98-105, ISSN 26663074