

Comparative Analysis of Hate Speech Detection Models on Brazilian Portuguese Data: Modified BERT vs. BERT vs. Standard Machine Learning Algorithms

Thiago Mei Chu¹, Leila Weitzel¹^a and Paulo Quaresma²^b

¹*Department of Computer Science, Fluminense Federal University, Rio das Ostras, Brazil*

²*Department of Informatics, University of Évora, Évora, Portugal*

Keywords: Hate Speech Detection, Machine Learning, BERT, Transformer.

Abstract: The Internet became the platform for debates and expression of personal opinions on various subjects. Social media have assumed an important role as a tool for interaction and communication between people. To understand this phenomenon, it is indispensable to detect and assess what characterizes hate speech and how harmful it can be to society. In this paper we present a comprehensive evaluation of Portuguese-BR hate speech identification based on BERT model and ML models as baseline. The BERT model achieves higher scores compared to the machine learning algorithms, indicating better overall performance in distinguishing between classes.

1 INTRODUCTION


The identification of online hate speech for research purposes is confronted with numerous challenges, from a methodological perspective – including definitions used to frame the issue, social and historical contexts, linguistic subtleties, the variety of online communities and forms of online hate speech (type of language, images, etc.) (UNESCO, 2021).


In the contemporary interconnected world, the potency of language cannot be underestimated. While the freedom of speech is a fundamental right, it entails the duty to utilize language in a manner that nurtures comprehension, respect, and concordance. Unfortunately, hate speech and offensive language have proliferated, posing a considerable threat to individuals and society at large. Their ramifications extend well beyond verbal expression, causing psychological and emotional harm, undermining social cohesion, threaten democracy and human rights, and creating widespread societal repercussions. In this context, Artificial Intelligence (AI) and Machine Learning (ML) play a strategic role in identifying and mitigating harmful content on social networks (Mondal et al., 2017; Mozafari et al., 2022; Watanabe et al., 2018).

In the context of widespread internet access and the extensive use of online social networks, the current communication paradigm is focused on sociability and socialization, with an emphasis on the social utilization of technology. This paradigm surpasses the traditional notion of communication as a mediation between two entities - sender and receiver - and instead expands this perspective to encompass multiple entities, including individuals, communities, and society. As a result, it gives rise to new forms of interaction within communication processes (Maia & Rezende, 2016).

During the 2018 Brazilian presidential election, a significant number of controversial and even unacceptable user comments began to surface. Brazil joins a growing list of countries where social media disinformation has been employed to manipulate real-world behavior. The campaign period was marred by political violence and the dissemination of election-related disinformation and hate speech on social media and messaging platforms. For our research, we chose to focus on X (Twitter) in Portuguese language.

As Twitter posts are primarily informal, analyzing this kind of text can present more significant challenges compared to formal texts (Bahrainian & Dengel, 2013).

^a <https://orcid.org/0000-0002-3090-3556>

^b <https://orcid.org/0000-0002-5086-059X>

In this context, the main goal of this research is to carry out a comparative analysis of hate speech detection. This is achieved through the employment of several processing strategies and models in a database in the Brazilian Portuguese language. The database comprises tweets gathered during the electoral period and the subsequent post-election phase for the Brazilian presidential election covering from 2018 to 2020. A total of 21,725 tweets were gathered, with 2,443 labelled as hate speech and 19,282 as non-hate speech (Weitzel et al., 2023).

2 HATE SPEECH IN SOCIAL MEDIA

Hate speech in social media refers to any form of communication that expresses hatred, prejudice, or intolerance towards an individual or group based on characteristics such as race, ethnicity, religion, sexual orientation, or gender identity. The uncontrolled spread of hate has far-reaching consequences, severely harming our society and causing damage to marginalized individuals or groups. Social media serves as one of the primary arenas for the dissemination of hate speech online. This harmful communication takes various forms, including derogatory language, threats, harassment, and incitement to violence. However, automatically detecting hate speech faces significant challenges. Social media posts often include paralinguistic signals (such as emoticons and hashtags) and frequently contain poorly written text. Additionally, the contextual nature of the task and the lack of consensus on what precisely constitutes hate speech make the task difficult even for humans. Furthermore, creating large labelled corpora for training such models is a complicated and resource-intensive process. To tackle these challenges, natural language processing (NLP) models based on deep neural networks have emerged. These models aim to automatically identify hate speech in social media data, contributing to the preservation of social cohesion, democracy, and human rights (Anjum & Katarya, 2024; Ermida, 2023; Guiora & Park, 2017; Maia & Rezende, 2016).

2.2 Hate Speech in Brazil

Hate speech in Brazil has distinctive features, mainly due to two factors: the complexity of the language and the way Brazilians express their emotions. These factors pose an additional challenge in classification

tasks. Hate speech in Brazil can manifest in various forms, reflecting the country's unique cultural context and linguistic nuances. Here are some examples of hate speech in Brazil: Racial Slurs: Offensive language targeting racial or ethnic groups based on skin color, ancestry, or nationality. These slurs perpetuate discrimination and prejudice; Homophobic Remarks: Brazil has a significant LGBTQ+ community, but unfortunately, hate speech against sexual minorities persists; Misogyny: Sexist language and misogyny are widespread. Women face derogatory comments, objectification, and threats online and offline and Political Attacks: Brazil's polarized political climate leads to hate speech against opposing parties, politicians, and their supporters. Such discourse undermines healthy democratic dialogue.

Two aspects must be taken into account in NLP tasks when dealing with texts in Brazilian Portuguese. The first aspect is that Brazilians have the habit of using swearword to express themselves freely. This characteristic imposes additional challenges in the classification task. Therefore, manual annotation becomes a critical factor.

The second aspect that contributes to the challenge of NLP tasks is the fact that English and Portuguese are two completely distinct languages in their formation. There are notable differences in grammar, vocabulary, and pronunciation between the two languages. One key difference is their grammatical structure, where English is considered more analytic, relying on word order and auxiliary verbs to convey meaning, while Portuguese is more synthetic, using inflections and grammatical markers to indicate relationships between words. Another significant difference lies in their vocabulary. Portuguese has a rich vocabulary with many words derived from Latin but also incorporates influences from indigenous languages and African languages. Portuguese has a more elaborate system of verb conjugations and grammatical genders, which can be challenging for non-native speakers to master. Additionally, Portuguese has a greater number of verb tenses and moods compared to English, adding to its complexity. The reason Portuguese is often considered more complex than English lies in its grammar (dos Santos, 1983; Roscoe-Bessa et al., 2016). Furthermore, there is a lack of linguistic resources for the Brazilian language. The literature offers a wealth of resources for the English language, and a significant number for the European Portuguese language. Brazilian Portuguese and European Portuguese, while sharing similarities, exhibit differences in morphosyntactic structure, phonetics,

and vocabulary. These languages serve linguistic communities that are not only geographically distant.

3 METHODOLOGY

This section outlines the fundamental principles employed in constructing the research methodology and delineates the stages of each process.

3.1 Data Gathering Process

In our preceding research, a comprehensive collection and processing of a tweet database was undertaken. The detailed methodology employed in this phase is extensively documented in (Weitzel et al., 2023). To facilitate the manual labeling of tweets, a lexicon of offensive language specific to the Portuguese language was developed. This manual labeling process involved the participation of seven native Portuguese speakers. The selection of these participants encompassed a broad age range, from 18 to 65 years old, to ensure the capture of the full spectrum of spoken language subtleties. As noted by Rodrigues (1981), language is a dynamic and continually evolving system of communication that serves to both shape and reflect the cultural identity, historical context, and societal developments of a specific speaker group.

To assess the agreement rate among the researchers, we computed Cohen's Kappa, a well-established interrater reliability score. A kappa value of 1 signifies complete agreement among the researchers, while a kappa value of 0.0 indicates no agreement. The achieved kappa values were approximately 0.5, suggesting a moderate level of agreement. As previously mentioned, a total of 21,725 tweets were gathered, with 2,443 labelled as hate speech and 19,282 as non-hate speech (Hidden for blind version, 2023).

In the realms of machine learning and data mining, dealing with class imbalance is a significant challenge, primarily due to the biased nature of the data towards the majority class. An unbalanced dataset occurs when one class (the majority class) significantly outweighs another (the minority class) in terms of examples. One of the main issues with unbalanced datasets is that machine learning models trained on such data tend to perform poorly in predicting the minority class. This is because the model is biased towards the majority class and may not learn enough about the minority class to make accurate predictions. As a result, the model may have high accuracy for the majority class but poor

performance for the minority class. To address these challenges, several techniques can be used to balance the dataset, such as resampling methods (oversampling the minority class or undersampling the majority class) (Rawat & Mishra, 2022). As a result, we applied the undersampling technique, resulting in a balanced dataset of 4,886 tweets.

3.2 BERT and Traditional ML Models

In this study, we compared the performance of four traditional machine learning models (Support Vector Classifier, Naive Bayes, Multilayer Perceptron, Logistic Regression) as baseline and BERT (Bidirectional Encoder Representations from Transformers) model, based on the *bert-base-portuguese-cased* which is available at (<https://huggingface.co>). It is pre-trained on a corpus of text in Portuguese and is cased, meaning it retains the case information of the input text. This model has been widely used for various natural language processing tasks, including text classification, named entity recognition, and question answering. Its key features are (Lin et al., 2022; Turner, 2024):

- **Bidirectional Context:** BERT can understand the context of a word based on its surrounding words in both directions, allowing it to capture complex linguistic patterns
- **Transformer Architecture:** BERT is based on the transformer architecture, which allows for parallel processing of words in a sequence, making it highly efficient for processing long sequences of text. The basic architecture of BERT consists of an encoder with multiple layers of self-attention mechanisms. Each layer refines the representation of the input text, allowing BERT to capture hierarchical and contextual information.

Initially, we used the standard BERT model (our first model), which means, without changing its architecture. Subsequently, we made some adjustments to this architecture in order to enhance the performance.

3.3 Fine Tuning

To the best of our knowledge, the BERT model, mainly, "bert-base-portuguese-cased" does not have automatic optimization techniques built into its architecture. Fine-tuning this model typically involves manually adjusting hyperparameters such as learning rate, batch size, and the number of training epochs based on empirical testing. Additionally, architectural modifications can be explored to improve performance, such as adjusting the attention

mechanism adding or removing layers. While there are no built-in automatic optimization techniques for this model, external tools and frameworks such as AutoKeras or hyperparameter optimization libraries cannot be used to automate the fine-tuning process. Hence, the entire fine-tuning process was conducted manually. We highlight the parameters that were adjusted during training:

- **Optimizer:** We experimented with AdamW, RMSprop, and Adam.
- **Learning Rate:** For instance, we explored values ranging from $2e-5$ to $3e-5$, commonly used in transformer models.
- **L2 Regularization Technique** was also employed for the weights and biases of the dense layers in a neural network model. L2 Regularization Setup was: `weight_decay = 0.01` that specifies the L2 regularization strength. The code iterates over each layer in the model to check if it is a dense layer. For each dense layer, L2 regularization is applied to the weights. If the layer has a bias term L2 regularization is also applied to the bias. L2 regularization helps prevent overfitting by adding a penalty term to the loss function that is proportional to the squared magnitude of the weights. This encourages the model to learn simpler patterns and reduces the likelihood of overfitting to the training data.
- **Tokenizer:** Tokenization is the process of breaking down a text into smaller units called tokens, which can be words, subwords, or characters. which is essential for processing text data in natural language processing (NLP) tasks. The BERT function is `BertTokenizer`, certain parameters, such as padding, truncation, and `max_length`, must to be specified. Padding ensures that all tokenized sequences are padded to the same length specified by `max_length`. Padding is necessary because neural networks typically require inputs of fixed length. `Truncation=True`: Specifies that sequences longer than `max_length` should be truncated to `max_length`. Truncation ensures that all sequences have the same length, which is important for efficient batch processing. `Max_length`: Specifies the maximum length of the tokenized sequences. This parameter is important because it controls the length of the input sequences fed into the model during training and inference. Setting an appropriate `max_length` helps balance the trade-off between computational efficiency and preserving important information in the input text. The `max_length` values tested ranged from 70 to 100. The maximum calculated value was 85 tokens per tweet. It is important to highlight that most tweets contain between 30 and 40 tokens.

- **Callback EarlyStopping:** it is a form of regularization used to avoid overfitting when training a learner with an iterative method, such as gradient descent. This technique ends training when a monitored metric has stopped improving. Early Stopping is a form of regularization technique that prevents overfitting of the model to the training data. Overfitting occurs when a model learns the training data too well, capturing noise along with the underlying pattern.
- **Batch_size:** batch size refers to the number of training examples utilized in one iteration. The values 16, 32 and 64 were tested.
- **Epoch:** is a term used in machine learning and indicates the number of passes of the entire training dataset the machine learning algorithm has completed. If the batch size is equal to the total dataset size, one epoch is equivalent to one update to the model. If the batch size is less than the total dataset size, one epoch will involve multiple updates to the model. The epoch was controlled by the early stopping function. It means that the number of epochs, or complete passes through the training dataset, was determined by the early stopping function. This function monitors a specified metric and stops training when that metric stops improving, effectively controlling the number of epochs.

3.4 Modified BERT Architecture

The chosen model architecture consists of the following key components:

- We remove the **activation function**, `bert_model.layers[-1].activation = None`
- **Input Layers:** Three input layers (`input_ids`, `attention_mask`, `token_type_ids`) are defined to receive input sequences for BERT; The model takes tokenized input text, consisting of `input_ids`, `attention_mask`, and `token_type_ids`.
- **attention_mask** = `Input (shape = (80,), dtype = 'int32')`. This line of code defines an input layer named `attention_mask` that expects sequences of length 80 with integer values. This layer can be used to pass an attention mask to the model, which can be used to indicate which parts of the input sequence should be attended to during processing. The attention mask is an optional argument in the BERT model. It's used when batching sequences together to indicate which tokens should be attended to, and which should not⁵. When `attention_mask == 1`, it indicates that attention is paid to the token. Forcing it to zero effectively makes the token invisible. So, while BERT does not

require an attention mask by default, it can be beneficial to include it when dealing with sequences of varying lengths or when you want to make certain tokens invisible to the model. If you don't provide an attention mask, the model will attend to all tokens in the sequence. Therefore, depending on your specific use case, you might need to add a line of code to handle the attention mask.

- **Dropout Layer:** Introducing dropout to prevent overfitting by randomly setting a fraction of input units to zero during training. A dropout layer with a dropout rate of 0.1 was applied to the BERT output to prevent overfitting.
- **Dense Layer:** A dense layer with 64 units and ReLU activation is added after the dropout layer.
- **Output Layer:** Finally, a dense layer with 1 unit and sigmoid activation is added as the output layer for binary classification tasks.

The bert_model is called with the input layers, which returns a sequence of hidden states. [0] is used to extract the output of the BERT model (i.e., the hidden states). The input_ids represent the tokenized input text, where each token is mapped to a unique integer ID. The attention_mask is used to indicate which tokens should be attended to and which should be ignored, helping the model focus on relevant parts of the input. The token_type_ids are used in tasks where inputs consist of two sequences (e.g., question answering), indicating which tokens belong to which sequence. The BERT model is used to generate embeddings for the input tokens, which are then passed through a dropout layer to prevent overfitting. The output of the dropout layer is then fed into a dense layer with 64 units and ReLU activation function, which helps in learning complex patterns in the data. Finally, the output is passed through another dense layer with a single unit and sigmoid activation function, which is suitable for binary classification tasks, producing a probability score for the positive class. The model is compiled using binary cross-entropy loss, which is well-suited for binary classification problems. The model is evaluated based on accuracy, which measures the proportion of correctly classified samples. We use binary cross-entropy loss, which is suitable for binary classification tasks, to measure the difference between predicted probabilities and actual labels.

4 EXPERIMENTAL RESULTS

The training of the BERT Network began with the standard model, without any modifications.

Throughout the training process, we were monitoring the val_loss, i.e., monitoring the loss function during the validation level. We evaluated the aforementioned parameters, including the learning rate, optimizers, batch size, and token count, in the standard model.

The EarlyStopping function stops the training process when the metric stops improving, or starts to worsen, for a certain number of epochs (defined by the patience parameter equal to 2). The parameter indicates that training should stop if the validation loss does not improve for two consecutive epochs. This was one of the more favourable results achieved in the various trainings conducted, Table 1. Although most models achieved relatively high performance on both the training and validation sets.

Examining Figure 1, it becomes apparent that the accuracy on the test set surpassed that of the training set. When the accuracy on the test set is higher than that on the training set, it can indicate several things. Firstly, it could suggest that the model is not generalizing well to unseen data, as it is performing better on the data it has already seen (training set) compared to new data (test set). A small difference may not be concerning and could be due to random fluctuations.

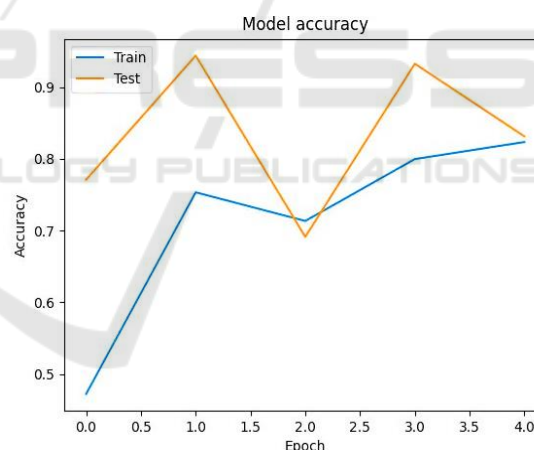


Figure 1: The plot diagram shows the overfitting during training and testing with Standard BERT.

Table 1: This is the most positive result achieved with standard BERT.

LOSS	ACCURACY	VAL_LOSS	VAL_ACC
0.437	0.524	0.24	0.788
0.203	0.775	0.192	0.558
0.127	0.284	0.199	0.41

The Figure 2 shows the training and testing accuracies of a binary text classification model using modified BERT over several epochs. Both the training and testing accuracies increase until around the 4th epoch, which is a good sign as it indicates that the model is learning from the training data and is able to generalize well to new, unseen data. However, after the 4th epoch, the training accuracy continues to increase while the testing accuracy plateaus. This divergence between the training and testing accuracies is a classic sign of overfitting. It suggests that the model is becoming too specialized to the training data and is losing its ability to generalize to new data.

Table 2: Modified BERT Results.

Bert_Metrics	Score
F1-score	0.932
Accuracy	0.973
Precision	0.974

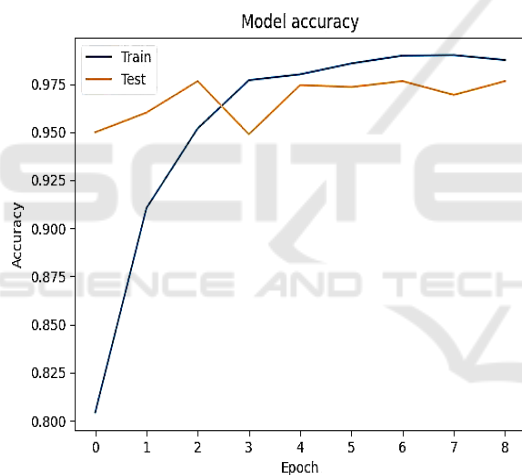


Figure 2: Modified BERT Model accuracy: train and test.

Figures 2 and 3 illustrate the accuracy and loss charts corresponding to the best-performing model, achieved with the following parameters:

- max_length = 80.
- batch_size = 32.
- Adam optimizer with a learning rate of 3e-5,
- dropout = 0.1.
- We add two dense layers. The first dense layer has 64 units and uses the ReLU activation function to capture complex patterns in the data. The second dense layer has a single unit with a sigmoid activation function, suitable for binary classification, producing a probability score for the positive class.
- Training stopped at epoch = 10.

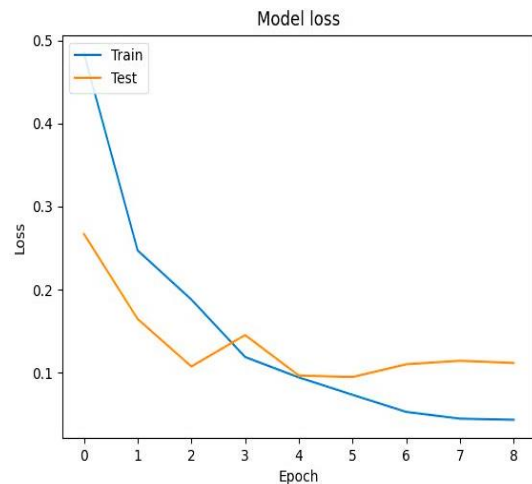


Figure 3: Modified BERT Model loss: train and test.

4.2 Standard Machine Learning Results Discussion

Table 3 presents the outcomes achieved from training the dataset using the above-mentioned machine learning algorithms.

Table 3: Standard ML Results.

METRICS	ACCURACY	PRECISION	RECALL	AUC	F1-SCORE
SVC	0.947	0.9807	0.907	0.947	0.945
NB	0.911	0.900	0.928	0.911	0.914
MLP	0.914	0.911	0.922	0.314	0.916
LR	0.934	0.974	0.893	0.934	0.932

The results obtained from the various machine learning algorithms provide valuable insights into their performance in classifying hate speech.

- **Support Vector Classifier (SVC)** shows the highest performance across all metrics among the four models. SVC appears to be the most effective model for this particular task. We observe a high level of accuracy at 94.7%, indicating its effectiveness in correctly classifying instances. The precision of 98.7% implies that when the SVC predicts a tweet as hateful, it is correct 98.7% of the time. However, the recall of 90.7% suggests that there is still room for improvement in identifying all hateful tweets, as some are being missed. The Area Under the Curve (AUC) score of 94.7% indicates a good overall performance, and the F1 score of 94.5% balances the precision and recall metrics.
- **Naive Bayes (NB)** model, we see a lower accuracy of 91.1% compared to the SVC. The precision of 90% and recall of 92.8% suggest that the NB model

is slightly less effective in correctly identifying hateful tweets, with a balanced F1 score of 91.4%. This suggests that while NB may make more mistakes overall, it is less likely to miss positive instances.

- **Multilayer Perceptron (MLP)** achieves an accuracy of 91.4%, similar to the NB model. The precision of 91.1% and recall of 92.2% indicate a balanced performance, resulting in an F1 score of 91.6%. This suggests that MLP and NB have similar performance, with MLP making slightly fewer mistakes.
- **Logistic Regression (LR)** model achieves an accuracy of 93.4%, with a high precision of 97.4% but a lower recall of 89.3% is the lowest among the four models. This suggests that while LR is generally accurate and makes few false positive errors, it is more likely to miss positive instances than the other models. The AUC score of 93.4% and F1 score of 93.2% demonstrate a good overall performance, although there is room for improvement in recall.

We can infer that SVC model demonstrates the best overall performance among the tested algorithms, with high accuracy, precision, and a balanced F1 score. The LR model also performs well but with slightly lower recall. The NB and MLP models show comparable performance, with the NB model having slightly lower accuracy but higher recall than the MLP model. Overall, these results highlight the importance of selecting the right machine learning algorithm for classifying hate speech, taking into account the trade-offs between accuracy, precision, recall, and other metrics. However, it is important to note that these results are specific to the particular dataset and task at hand. Different tasks or datasets might yield different results. Therefore, it's always recommended to experiment with various algorithms and tune their hyperparameter.

4.3 BERT Models and Standard ML Results Discussion

In the field of deep learning, the performance of a model is often a function of various hyperparameters, one of which is `max_length`. This parameter typically denotes the maximum length of the token sequences that the model processes. In the context of BERT and similar transformer models, `max_length` can significantly impact both the computational efficiency and the performance of the model. Interestingly, it is not always the case that a larger

`max_length` leads to better model performance. As observed in some scenarios, a model can achieve optimal performance with a `max_length` value that is less than the actual maximum length of the token sequences in the dataset. In the given scenario, the actual `max_length` was 85 tokens, but the model achieved the best results with a `max_length` of 80 tokens. This phenomenon can be attributed to several factors:

- **Computational Efficiency:** A smaller `max_length` reduces the computational complexity of the model, as there are fewer tokens to process in each sequence. This can lead to faster training times and less memory usage.
- **Noise Reduction:** By limiting the `max_length`, the model might be focusing on the most relevant parts of the sequence and ignoring potential noise in the longer tail of the sequence.
- **Regularization Effect:** A smaller `max_length` can also act as a form of regularization, preventing the model from overfitting to the training data by limiting its capacity to memorize long sequences.

However, it is important to note that this does not imply that reducing `max_length` will always improve performance. The optimal `max_length` is problem-specific and should be determined through careful experimentation. While longer sequences can provide more context for the model, they also introduce challenges related to computational efficiency and overfitting. Therefore, finding the right balance is crucial for achieving optimal model performance.

In the original BERT paper by Google, a batch size of 32 was used for fine-tuning the model on specific tasks. This batch size was found to be a good balance between computational efficiency and model performance. However, it is important to note that this does not mean that a batch size of 32 will always be the optimal choice. The optimal batch size can vary depending on the specifics of the task and the computational resources available. Larger batch sizes can lead to faster training, but also require more memory and may lead to less accurate models. On the other hand, smaller batch sizes can lead to more accurate models, but require more iterations to train. The optimal batch size should be determined through careful experimentation considering the specific task, the size of the training data, and the computational resources.

The modified BERT architecture, as described, demonstrates a compelling performance on the given task, as evident from the results. The model shows a consistent decrease in loss and an increase in

accuracy across epochs, indicating successful learning and generalization capabilities.

One notable aspect is the model's ability to achieve high accuracy on both the training and validation sets, reaching up to 98.9% and 97.6%, respectively. This suggests that the model is effectively capturing the underlying patterns in the data and is not overfitting, as the validation accuracy remains close to the training accuracy throughout training.

The loss values also show a decreasing trend, which is expected as the model learns to minimize its error. The final loss values on the training and validation sets are quite low, indicating that the model is able to make accurate predictions with high confidence.

Overall, these results suggest that the modified BERT architecture, with the added dense layers and dropout, is effective for the given classification task. The model demonstrates strong learning and generalization capabilities, achieving high accuracy and low loss on both the training and validation sets. This highlights the potential of BERT-based models for similar natural language processing tasks, showcasing their ability to learn complex patterns and achieve high performance.

4.4 Comparing Results

Starting with the BERT model, we see that it achieves a maximum accuracy of 99.0% and a minimum accuracy of 80.4%. This indicates that the BERT model consistently outperforms the machine learning algorithms in terms of accuracy. The BERT model also achieves higher scores compared to the machine learning algorithms, indicating better overall performance in distinguishing between classes. However, when comparing precision, recall, and F1 score, we see some variation. The BERT model achieves high precision, recall, and F1 scores, indicating its effectiveness in correctly classifying both positive and negative instances.

On the other hand, the machine learning algorithms show a wider range of precision, recall, and F1 scores, with some models performing better than others in different metrics. While the BERT model consistently outperforms the machine learning algorithms in terms of accuracy, there are differences in precision, recall, and F1 score. This suggests that while the BERT model may be more accurate overall, there may be trade-offs in terms of precision and recall compared to the machine learning algorithms.

Overall, the choice between the BERT model and machine learning algorithms depends on the specific

requirements of the task and the importance of different evaluation metrics. While the BERT model shows promising results, there is still room for improvement and further research to fully leverage its potential in natural language processing tasks.

5 CONCLUSION AND FUTURE WORK

The Internet became the platform for debates and expressions of personal opinions on various subjects. Social media have assumed an important role as a tool for interaction and communication between people. The advent of social media has revolutionized political discourse, providing a platform for citizens to express their views and engage in political debates. However, this freedom of expression has also given rise to a darker phenomenon: hate speech. This has been particularly evident in Brazil during the presidential elections of 2018 and 2022.

In 2018, Brazil's presidential election marked a significant turning point in the country's political landscape. It was during this period that the country witnessed a surge in hate speech. The campaign was marred by instances of political violence, the spread of misinformation, and a proliferation of hate speech and offensive content on social media platforms. The tone of online conversations became noticeably more aggressive, with hostility and hate speech flourishing. Most of this online hate speech targeted politicians and minority groups, focusing on their race, religion, and/or sexual orientation. Fast forward to 2022, the situation did not seem to improve. The presidential election that year was once again characterized by a high level of hate speech. The reasons for this are manifold, ranging from the polarized political climate and the rise of populist rhetoric to the misuse of social media platforms and the challenges associated with moderating online content. The persistence of hate speech during these election periods raises serious concerns about the health of Brazil's democratic process. It not only undermines the principles of respect and tolerance that are fundamental to any democratic society but also threatens to further polarize the country's already divided political landscape.

To understand this phenomenon, it is indispensable to detect and assess what characterizes hate speech and how harmful it can be to society. In this paper we present a comprehensive evaluation of Portuguese-BR hate speech identification based on BERT model and ML models as baseline.

The comparison between the BERT model and the ML algorithms provides valuable insights into their performance on the classification task. While the BERT model demonstrates superior accuracy, there are areas where it can be improved and future work can be focused. One area for improvement is the fine-tuning of the BERT model. Fine-tuning involves adjusting the hyperparameters and architecture of the model to better fit the specific task at hand.

Another aspect to consider is the use of domain-specific pre-training or transfer learning. Fine-tuning BERT on a dataset that is more closely related to the classification task, or using a pre-trained model that has been specifically trained on a similar domain, could lead to better performance. Furthermore, ensemble methods could be explored to combine the predictions of multiple models, including both the BERT model and the machine learning algorithms. Ensemble methods have been shown to improve performance by leveraging the strengths of different models.

In terms of future work, one direction could be to explore multi-task learning with the BERT model. Multi-task learning involves training the model on multiple related tasks simultaneously, which could lead to improved performance on each individual task. Additionally, investigating the interpretability of the BERT model's predictions could provide valuable insights into its decision-making process. Techniques such as attention mapping and feature visualization could be employed to understand which parts of the input are most influential in the model's predictions.

REFERENCES

- Anjum, & Katarya, R. (2024). Hate speech, toxicity detection in online social media: a recent survey of state of the art and opportunities. *International Journal of Information Security*, 23(1), 577-608. <https://doi.org/10.1007/s10207-023-00755-2>
- Bahrainian, S., & Dengel, A. (2013, 17-20 Nov. 2013). Sentiment Analysis Using Sentiment Features. 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT).
- dos Santos, A. S. (1983). *Guia prático de tradução inglesa: comparação semântica e estilística entre os cognatos de sentido diferente em inglês e em português*. Editora Cultrix. <https://books.google.com.br/books?id=Ot0uAAAAYAAJ>
- Ermida, I. (2023). Distinguishing Online Hate Speech from Aggressive Speech: A Five-Factor Annotation Model. In I. Ermida (Ed.), *Hate Speech in Social Media: Linguistic Approaches* (pp. 35-75). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-38248-2_2
- Guiora, A., & Park, E. A. (2017). Hate Speech on Social Media. *Philosophia*, 45(3), 957-971. <https://doi.org/10.1007/s11406-017-9858-4>
- Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, 3, 111-132. <https://doi.org/https://doi.org/10.1016/j.aiopen.2022.10.001>
- Maia, R. C. M., & Rezende, T. A. S. (2016). Respect and Disrespect in Deliberation Across the Networked Media Environment: Examining Multiple Paths of Political Talk. *21(2 %J J. Comp.-Med. Commun.)*, 121-139. <https://doi.org/10.1111/jcc4.12155>
- Mondal, M., Silva, L. A. j., & Benevenuto, F. c. (2017). A measurement study of hate speech in social media. Proceedings of the 28th acm conference on hypertext and social media,
- Mozafari, M., Farahbakhsh, R., & Crespi, N. (2022). Cross-Lingual Few-Shot Hate Speech and Offensive Language Detection using Meta Learning. *IEEE Access*, 1-1. <https://doi.org/10.1109/ACCESS.2022.3147588>
- Rawat, S. S., & Mishra, A. K. (2022). Review of Methods for Handling Class-Imbalanced in Classification Problems. <https://doi.org/10.48550/ARXIV.2211.05456>
- Roscoe-Bessa, C., Pessoa, M. N., & Dias, I. C. B. (2016). Algumas diferenças comunicativas entre o português e o inglês. *Cadernos de Tradução*, 36(2), 91. <https://doi.org/10.5007/2175-7968.2016v36n2p91>
- Turner, R. E. (2024). An Introduction to Transformers. In: arXiv.
- UNESCO. (2021). *Addressing hate speech on social media: contemporary challenges*. Retrieved from <https://unesdoc.unesco.org/ark:/48223/pf0000379177>
- Watanabe, H., Bouazizi, M., & Ohtsuki, T. (2018). Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. *IEEE Access*, 6, 13825-13835. <https://doi.org/10.1109/ACCESS.2018.2806394>
- Weitzel, L., Daroz, T. H., Cunha, L. P., Helde, R. V., & Morais, L. M. d. (2023, 20-23 June 2023). Investigating Deep Learning Approaches for Hate Speech Detection in Social Media: Portuguese-BR tweets. 18th Iberian Conference on Information Systems and Technologies (CISTI), Aveiro, PT.