Natural Language Content Evaluation System For Multiclass Detection of Hate Speech in Tweets Using Transformers

Duván Andres Marrugo-Tobón[†], Juan Carlos. Martinez-Santos and Edwin Puertas

Universidad Tecnologíca de Bolívar, Faculty of Engineering, Cartagena de Indias 17013001, Colombia

Abstract

In natural language processing, accurate categorization of tweets, including detecting hate speech, plays a pivotal role in efficient information organization and analysis. This paper presents a Natural Language Contents Evaluation System specifically tailored for multi-class tweet categorization, focusing on hate speech detection. Our system enhances classification accuracy and efficiency by harnessing the power of Transformers, namely BERT and DistilBERT. By leveraging feature extraction techniques, we capture pertinent information from tweets, enabling practical analysis, categorization, and identification of hate speech instances. During training, we also tackle imbalanced corpora by employing techniques to ensure fair representation of different tweet categories, including hate speech. Our system achieves impressive accuracy through extensive training of 95%, showcasing Transformers' effectiveness in comprehending and categorizing tweets, including identifying hate speech. Furthermore, our system maintains a good accuracy during testing of 83%, highlighting the robustness and generalizability of the trained models for hate speech detection. This system contributes to advancing automated tweet categorization, specifically in hate speech detection, providing a reliable and efficient solution for organizing and analyzing diverse tweet datasets.

Keywords

Natural language processing, Hate speech detection, Transformers, DistilBERT, BERT, Feature extraction, Tweet categorization

1. Introduction

With the exponential growth of social media platforms, particularly Twitter, user-generated content has skyrocketed, making effective categorization and analysis of tweets increasingly challenging [1]. The categorization of tweets into meaningful and relevant categories is essential for various applications such as sentiment analysis, trend detection, and information retrieval.

IberLEF 2023, September 2023, Jaén, Spain

[*] Corresponding author.

[†] These authors contributed equally.

marrugod@utb.edu.co (D. A. Marrugo-Tobón); jcmartinezs@utb.edu.co (J. Carlos. Martinez-Santos); epuerta@utb.edu.co (E. Puertas)

https://github.com/DuvanMarrugoT (D. A. Marrugo-Tobón); https://www.utb.edu.co/profesores/juan-carlos-martinez-santos-2/ (J. Carlos. Martinez-Santos); https://github.com/EdwinPuertas (E. Puertas)

0009-0002-9537-0209 (D. A. Marrugo-Tobón); 0000-0003-2755-0718 (J. Carlos. Martinez-Santos); 0000-0002-0758-1851 (E. Puertas)

CEUR Workshop Proceedings (CEUR-WS.org)

However, traditional rule-based approaches and simple machine-learning techniques often need to catch up to capturing tweets' complex linguistic patterns and nuances.

Social media platforms have become a prominent medium for expressing opinions, sharing information, and fostering social connections in the contemporary digital age. However, the general nature of online communication has also led to the proliferation of hate speech, posing significant challenges to maintaining a respectful and inclusive online environment [2]. Therefore, recognizing and effectively addressing hate speech is crucial to promoting online safety, combating discrimination, and ensuring positive user experiences [3].

The detection and classification of hate speech in tweets have gained considerable attention due to this social media platform's brevity and widespread use. Traditional approaches for hate speech detection often relied on handcrafted features and rule-based systems, but these methods needed to capture the nuances and evolving nature of hate speech-language. With the recent advancements in natural language processing (NLP), specifically the development of transformer-based models, more sophisticated and accurate techniques for hate speech detection have emerged [4].

To address these challenges, recent advancements in natural language processing (NLP) have focused on leveraging deep learning models, particularly Transformers, for tweet categorization [5]. Transformers, a class of deep learning architectures, have revolutionized NLP tasks by capturing contextual information and semantic relationships in an unparalleled manner. Among the popular Transformer models, BERT (Bidirectional Encoder Representations from Transformers) and DistilBERT (a distilled version of BERT) have emerged as powerful tools for text classification tasks [6].

Transformers have been widely employed in the context of tweet classification to improve accuracy and efficiency. However, to further enhance the performance of tweet categorization systems, it is crucial to consider feature extraction techniques that capture relevant information from tweets[7]. There are various feature extraction methods explored in the literature, including the use of character n-grams [8], word embeddings such as Word2Vec [9], and deep learning-based approaches like Convolutional Neural Networks (CNNs) [10]. These techniques enable the models to understand the nuanced characteristics of tweets and improve classification accuracy.

Furthermore, we must address the imbalanced corpora in tweet datasets to ensure fair representation of different categories during training [11]. Oversampling and undersampling are two commonly used strategies to balance the corpus. These techniques aim to create a balanced distribution of tweet categories, providing equal opportunities for the model to learn from all classes [12]. Other advanced approaches, such as Synthetic Minority Over-sampling Technique (SMOTE) [13] and Adaptive Synthetic Sampling (ADASYN) [14], have also been proposed to generate synthetic samples or adaptively adjust the sampling rate to achieve better corpus balance.

By combining the power of Transformers, effective feature extraction techniques, and corpus balancing strategies [15], we aim to enhance the accuracy and robustness of tweet classification systems, ultimately enabling more precise information retrieval and analysis in the realm of social media data [16]. These combined techniques allow the models to capture the contextual information and semantic relationships in tweets, leverage informative features extracted from tweet content, and ensure fair representation of different tweet categories during training. The integration of these components forms a comprehensive framework that addresses the challenges posed by imbalanced corpora and enhances the overall performance of tweet classification systems.

This paper explores Transformers' effectiveness, specifically BERT and DistilBERT, in identifying different types of hate speech in tweets using transformer-based models, by classifying hate speech into distinct categories, such as homophobia intolerance. The contributions of this work include the application of cutting-edge Transformers for the categorization of tweets, the exploration of feature extraction techniques to enhance classification accuracy, and a comprehensive evaluation of the proposed approach using a real-world dataset. This work is as follows: First, it overviews the related work on text categorization. Next, it describes the methodology implemented for classification models. Then, it presents experimental validation. Finally, it summarizes conclusions and future works.

2. Methodology

2.1. Dataset Description

The Hate Speech Detection track, organized by the Grupo de Ingeniería Lingüística at the Universidad Nacional Autónoma de México, focuses on classifying tweets in Mexican Spanish for LGBT+phobic content. The dataset comprises tweets from 2012 to 2022, providing a decade-long temporal span for analysis [17].

Participants in this track used a corpus of Mexican Spanish tweets. They aim to develop classification models that accurately determine whether a given tweet exhibits LGBT+phobic content. The classification categories include LGBT+phobic (P), not LGBT+phobic (NP), or not LGBT+related (NA), enabling a multi-class classification task. For example, the baseline required for the use of a Transformer was 75%. In turn, the dataset provides three columns of information: Id, content, and label. As a result, the corpus needs to be more balanced. For example, 62% are the NP class, 25% are the NA class, and 12% are the P class.

This competition offers an opportunity to delve into the challenges of hate speech detection within the context of Mexican Spanish tweets [17]. Participants can explore the linguistic nuances, evolving trends, and temporal variations in LGBT+phobic content over the ten years, facilitating a comprehensive understanding of online hate speech dynamics.

2.2. Classification Process

In Fig. 1, we illustrate the classification process. The dataset is read and stored in a Data-Frame with 2-column labels. We preprocess the data by removing empty words, URLs, punctuation marks, and special characters and converting them to lowercase. We also did data balancing. Furthermore, we conduct exploratory analysis to extract keywords, acronyms, and abbreviations, utilizing n-grams for feature selection.

Feature extraction is crucial in representing sentences or documents by assigning a probability of occurrence to words. We trained the system using 80% of the data. We used the remaining 20% for verification. Evaluation metrics, including F1 score, accuracy, precision, and recall, were utilized to identify the best-performing method.

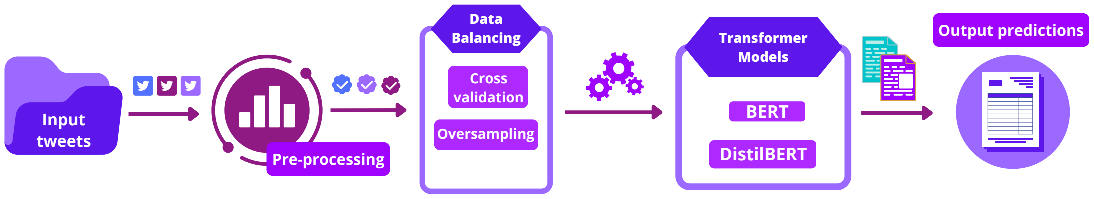


Figure 1: Transformer framework for tweet classification.

2.3. Pre-Processing

Input data for natural language tasks such as text classification consists of unstructured text. In contrast to other types of data, such as images or time series, textual information does not have an intrinsic numerical representation. Therefore, before entering it into a classifier, it has to be represented in a suitable feature domain. Therefore, preprocessing procedures are fundamental since, without them, there is no basis for feature extraction or classification algorithms [18].

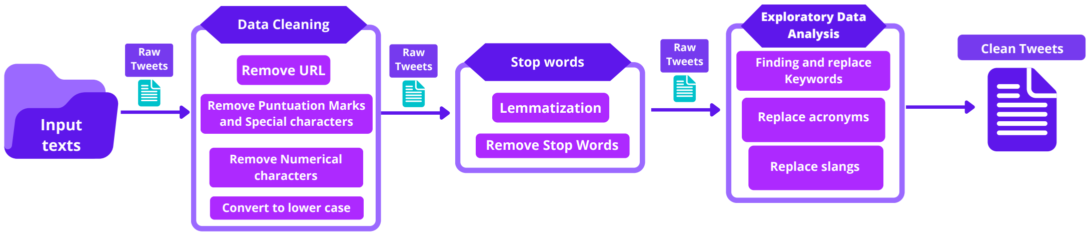


Figure 2: Data cleaning pipeline.

2.3.1. Data Cleaning

Data cleaning of tweets involves several steps to preprocess the text and remove unnecessary or irrelevant information, as shown in Fig. 2. The process includes removing special characters, punctuation marks, URLs, and mentions and converting the text into individual words. As well as removing empty words to reduce noise, hashtags, and emoticons were treated depending on the objectives of the analysis. In addition, text normalization is performed by converting the text to lowercase letters, processing abbreviations and contractions, and removing or substituting numbers.

2.3.2. Stopword

When working with Twitter data in Spanish, it is common to apply stopwords to remove words that do not contribute significantly to the text's overall meaning. These stopwords include articles like "el," "la," "los," and "las," pronouns such as "yo," "tú," and "él," as well as prepositions and conjunctions like "de," "en," "con," and "por." Additionally, common verbs like "ser," "estar," and "tener," along with adjectives and adverbs such as "bueno," "malo," "grande," and "poco," are often included in the stopwords list. Interjections like "ah," "eh," and "oh" are also considered as stopwords. Removing these stopwords from the Twitter text can shift the focus towards more meaningful content, allowing for better analysis and understanding of the data.

2.3.3. Exploratory Data Analysis

The text was normalized at this stage by converting it to lowercase and handling slang words and abbreviations identified based on [19]. Visualizing the content of a text document is a crucial task in text mining. However, many text visualizations indirectly represent the text by showing linguistic model outputs such as word count, character length, and word sequences. In our analysis, we initially conducted two types of analysis to explore the word relationships within each tweet:

- **Univariate Analysis:** This analysis shows that words like 'wuebonas'→ *'estupidas'*, 'ptm' → *'puta madre'* etc. These can be replaced by the expanded meaning or removed as needed. In our case, it increased the accuracy of the models by 0.03% compared to when we had it.
- **Bivariate Analysis:** Bigram and Trigram analysis was performed to explore word relationships within each tweet. Depending on the application, this analysis examined phonemes, syllables, letters, words, or base pairs. In addition, it aimed to identify any relationships with words that were not eliminated in previous phases, potentially impacting the subsequent classification process. The exploratory analysis revealed modified words, such as "Abbreviations" being transformed into "ntpsdv" and "no te pases de verga".

Thanks to these analyses, we can see which words affect the classifier's performance that we can eliminate. For example, words such as acronyms, abbreviations, grammatical errors, or slang ('vato', 'joto', 'wey', 'machin', etc) can be removed or replaced within the corpus.

2.3.4. Data Balancing

In the multiclass classification task, where the categories are LGBT+phobic (P), non-LGBT+phobic (NP), or non-LGBT+related (NA), data balancing is crucial due to the initial imbalance of classes in the dataset. Therefore, cross-validation and oversampling techniques were applied to address the dataset's initial class imbalance for the multiclass classification task.

We initially used the oversampling technique to increase the representation of the minority class (P) by generating synthetic samples. In this case, we used the popular oversampling technique Synthetic Minority Oversampling Technique (SMOTE). To avoid overfitting, we performed over-sampling. Initially, we performed a random oversampling with an algorithm randomness control (Random State) of 42. Then, we performed cross-validation by random permutation (Shuffle Split) to extract the training dataset equivalent to 80% and the test dataset comparable to 20% of the total data.

By combining cross-validation and oversampling, we can ensure that the model is trained on balanced data and evaluate its performance robustly. It helps mitigate the impact of class imbalance and improves the model's ability to accurately classify instances into all three classes: LGBT+phobic (P), non-LGBT+phobic (NP), and non-LGBT+ related (NA).

2.4. Transformers

Architecture

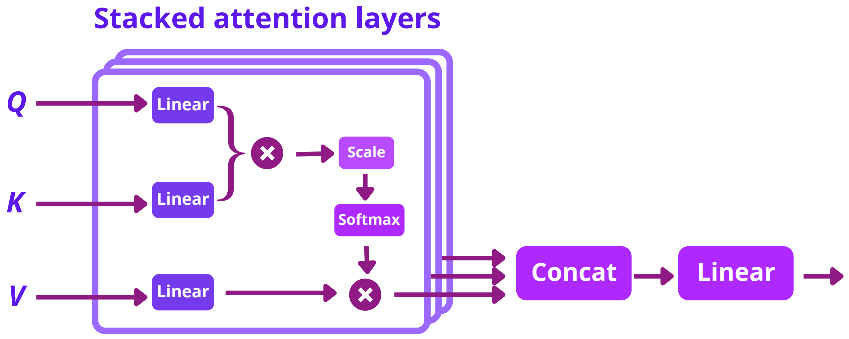


Figure 3: The multi-head attention layer used in the Transformer architecture.

A bag of tokens is given without any ordering pattern to initialize the transformers. Then, the transformer relies on a "self-attention" mechanism to learn the dependencies between the tokens. In addition, a particular encoding step is performed before the first layer of the encoder to ensure that embeddings of the same word appearing in a different position in the sentence will have another representation. This step is called positional encoding, and its purpose is to inject information about the relative positioning between words, which we would otherwise lose. The critical component of this architecture is the self-attention layer, which intuitively allows the encoder to look at other words in the input sentence whenever processing one of its words. Stacking multiple layers of this type creates multi-head attention (MHA) layer, as shown in Fig. 3. Then, we condensed the individual outputs into a single matrix by concatenating the head outputs and passing the result through a linear layer.

- **Encoder:** In the encoding part, the input embeddings are multiplied by three separate weight matrices, as indicated in Equation 1, Q (queries), K (keys), and V (values), to generate different word representations.

$$Q = X \cdot W_Q K = X \cdot W_K V = X \cdot W_V \tag{1}$$

W_Q, W_K, $W_V \in \Re^{dim \times d_k}$ are the learned weight matrices. Eventually, we obtain the representation of each word by multiplying the scaled term with the V matrix containing the input representation. We define this operation in Equation 2.

$$Z = Attention(Q, K, V) = softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{2}$$

- **Decoder:** During the decoding phase, every decoder layer receives the output of the encoder (the K and V matrices) and the output of the previous decoder layer. Additionally, we modified the self-attention layers into what we defined as "Masked" self-attention layers. The masked MH self-attention layer ensures the use of only the self-attention scores. We do it by adding a factor M to the word embeddings in Equation 3. We set M to -inf for masked positions and 0 otherwise.

$$Z = softmax(\frac{Q \cdot K^T + M}{\sqrt{d_k}}) \cdot V \tag{3}$$

- **Preprocessing:** For performing the preprocessing, we should note that the two proposed models, BERT and DistilBert, based on deep neural network architectures, include similar steps for removing special characters, lemmatization, and stop word removal. In addition, tokenized documents are truncated or padded with a given number of tokens to ensure that the model receives uniformly sized input samples (i.e., with the same number of tokens).

As mentioned above, we will develop the problem using pre-trained BERT and DistilBERT for automatic tweet categorization and test different optimizer methods. Table 1 shows the parameters used for each architecture.

BERT Architecture

The architecture consists of a stacked coding layer of the transformer [5]. BERT is composed of two main steps: pre-training and tuning. During pre-training, there are two unsupervised tasks to train BERT on a sizable unlabeled corpus: masked language modeling (MLM) and next sentence prediction (NSP) to produce a pre-trained model. Then, for fitting, the model is initialized with the pre-trained parameters, and all parameters are fit using labeled data for specific tasks such as classification.

We select a BERT-base model containing an encoder with 12 transformer blocks, 12 self-attenuating heads, and a hidden size 768. The network takes input from a sequence of no more than 512 tokens and outputs the sequence representation. The series has one or two segments in which the first token of the line is always [CLS], which contains the particular classification embedding. Then, we use another unique token [SEP] to separate the segments. This study applies the "BERT-base-uncased" model as the base model. In the base model, we use the tokenizer of the "BERT-base-uncased" model and the fine-tuned BERT architecture for the classification task. A simple SoftMax classifier is added to the model's top to predict the probability of label c. W is the task-specific parameter matrix, and n is a tweets' category.

DistilBERT Architecture

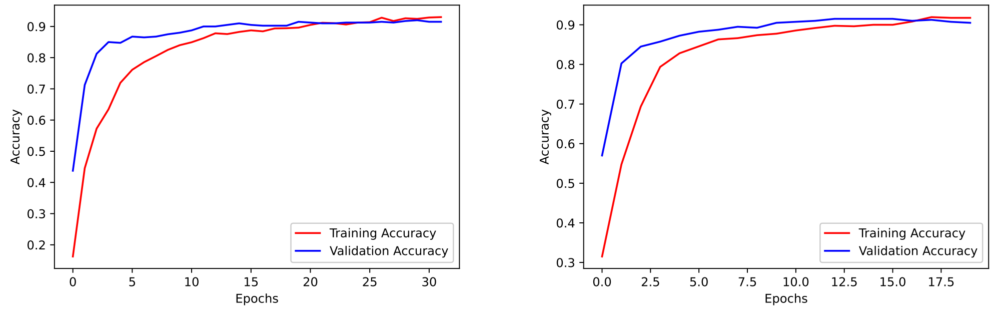
DistilBERT [20] is a "distilled" version of BERT, which is smaller and faster than BERT and also protects the accuracy of BERT. Therefore, it is safe to say that DistilBERT is a smaller transformer model than BERT. This model includes six layers, 768 dimensions, and 12 heads with 66 million parameters. We conducted experiments with DistilBERT to augment the results obtained with the "BERT-base-uncased-model". First, the "DistilBERT-base-uncased-model" is used as a pre-trained model. The final proposed architecture consists of DistilBERT [20] with dropout and linear layers on top of the DistilBERT. Next, we show the general architecture of the model. A softmax classifier is used on top of the linear layer to predict the probability of the n tweets category.

3. Experimental Results

Table 1
Hyperparameters used in both models

BERT		DistilBERT	
vocab size	128000	vocab size	128000
hidden size	768	hidden size	768
num hidden layers	12	num hidden layers	6
num attention heads	12	num attention heads	12
intermediate size	3072	intermediate size	3072
hidden dropout prob	0.1	hidden dropout prob	0.1
attention probs	0.1	attention probs	0.1
dropout prob	0.1	dropout prob	0.1
max position embeddings	512	Seq classify dropout	0.2
type vocab size	2	type vocab size	2
initializer range	0.02	initializer range	0.02
epoch	32	epoch	20
Optimizer	Adam Optimizer	Optimizer	Adamax Optimizer

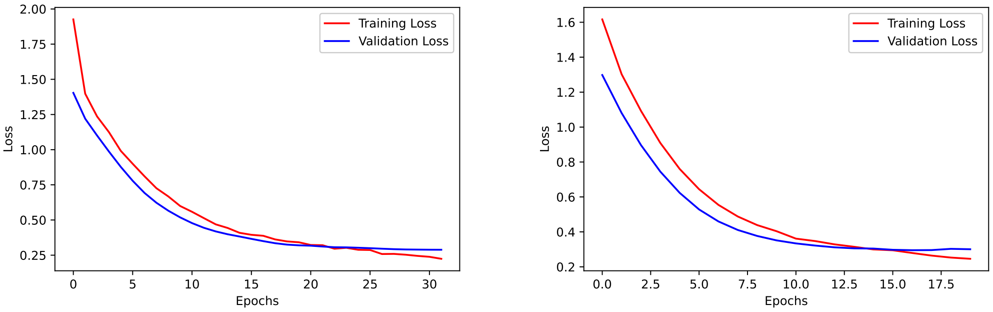
We initially trained the model designed with 80% of the data the competition provided. The construction of the BERT and DistilBERT models described before allowed us to obtain two models with an accuracy of 0.945 and 0.932. Since we performed both the analysis and training, accuracy and loss are the primary metrics. In Fig. 4 and 5, it is possible to observe the accuracy behavior for each epoch. Each model has a different number of epochs used in training. For the BERT model, there were 35, while for DistilBERT, 20. Although the accuracy obtained by the BERT model is higher than that of DistilBERT, it is worth considering how many epochs each one reaches the maximum level. Using an EarlyStopping, with patience=10, we obtained this information, which Table 2 shows. Based on the percentage of the duty cycle it takes to complete its maximum accuracy, DistilBERT does represent the most efficient model since it achieves similar accuracy to BERT in a shorter duty cycle. However, it is noteworthy to comment that



(a) Accuracy BERT model with 32 spochs.

(b) Accuracy DistilBERT model with 20 spochs.

Figure 4: Training and validation accuracy.



(a) Loss BERT model with 32 spochs.

(b) Loss DistilBERT model with 20 spochs.

Figure 5: Training and validation loss.

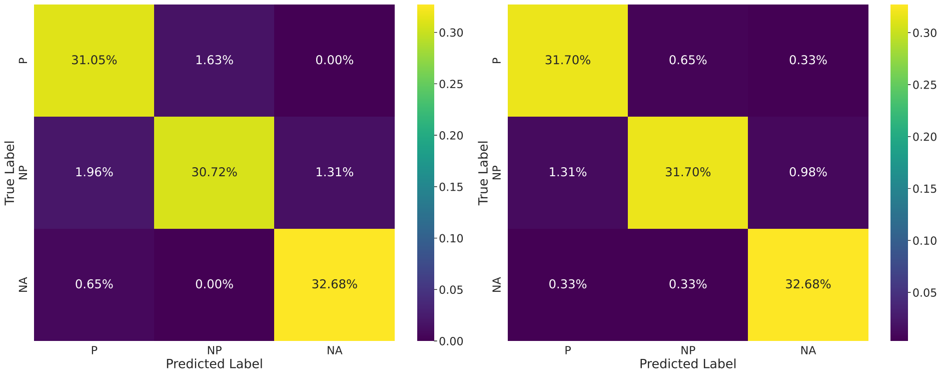
despite having the highest accuracy.

Table 2
Efficiency comparison measured over the duty cycle.

Model	Epoch	Loss	Accuracy	F1-Score	Duty cycle
BERT	25/32	0.2975	0.945	0.931	94.2 %
DistilBERT	13/20	0.3101	0.932	0.924	74 %

In Fig. 6(a), we can also see that despite not having equal or more significant precession than BERT, the DistilBERT model predicts the P and NA categories much better, with a minimum percentage error value of 2.25% concerning the data set. At the same time, it manages to be much more effective with prediction among the NP, which has a more significant number of data than the other classes, NA and P.

Finally, we tested the incoming model with a dataset provided without labels. The results



(a) BERT model

(b) DistilBERT model

Figure 6: Transformer model confusion matrices for tweets classification.

obtained respectively reach 82.4% and 81.3% accuracy.

4. Conclusions

In conclusion, our objective was to find an optimal model for the multiclass hate speech detection problem, explicitly focusing on classifying tweets in Mexican Spanish for LGBT+phobic content using the dataset provided by the Hate Speech Detection track organized by the Grupo de Ingeniería Lingüística at the Universidad Nacional Autónoma de México.

Firstly, although the DistilBERT model better predicted the P and NA categories, there is room for further improvement in accurately predicting the NP category. Fine-tuning the model or exploring different approaches tailored explicitly for handling imbalanced data could help address this issue. Additionally, while the BERT and DistilBERT models achieved high accuracies of 94.5% and 93.2%, respectively, there is potential for enhancing their performance even further. Experimenting with different hyperparameters and model architectures or incorporating additional contextual information could improve classification results.

We compared the results obtained using two transforms on the same corpus. The incoming model was then tested on a dataset without labels, achieving accuracies of 82.4% and 81.3%, respectively. Accuracy and loss were the primary metrics used for evaluation. Comparing the two transform-based models, we found that DistilBERT outperformed BERT in speed and accuracy, making it the preferred choice for classification. With an accuracy of 93.2%, it demonstrated better overall performance among the proposed models.

In summary, future work should address the remaining challenges in accurately predicting the NP category, refining the model's performance through advanced techniques, and broadening its language capabilities to encompass a more diverse range of languages. By continuing to iterate and improve upon the existing models, we can advance the field of hate speech detection and contribute to developing more robust and inclusive language processing solutions.

References

[1] H. Kim, Y.-S. Jeong, Sentiment classification using convolutional neural networks, Applied Sciences 9 (2019). URL: https://www.mdpi.com/2076-3417/9/11/2347. doi:10.3390/app9112347.

[2] M. Galas, Chapter 11 - experimental computational simulation environments for big data analytic in social sciences, in: V. Govindaraju, V. V. Raghavan, C. Rao (Eds.), Big Data Analytics, volume 33 of *Handbook of Statistics*, Elsevier, 2015, pp. 259–277. URL: https://www.sciencedirect.com/science/article/pii/B9780444634924000113. doi:https://doi.org/10.1016/B978-0-444-63492-4.00011-3.

[3] S. Abro, S. Shaikh, Z. H. Khand, Z. Ali, S. Khan, G. Mujtaba, Automatic hate speech detection using machine learning: A comparative study, International Journal of Advanced Computer Science and Applications 11 (2020). URL: http://dx.doi.org/10.14569/IJACSA.2020.0110861. doi:10.14569/IJACSA.2020.0110861.

[4] F. Alkomah, X. Ma, A literature review of textual hate speech detection methods and datasets, Information 13 (2022). URL: https://www.mdpi.com/2078-2489/13/6/273. doi:10.3390/info13060273.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[6] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. arXiv:1910.01108.

[7] H. A. Madni, M. Umer, N. Abuzinadah, Y.-C. Hu, O. Saidani, S. Alsubai, M. Hamdi, I. Ashraf, Improving sentiment prediction of textual tweets using feature fusion and deep machine ensemble model, Electronics 12 (2023). URL: https://www.mdpi.com/2079-9292/12/6/1302. doi:10.3390/electronics12061302.

[8] K. L. Tan, C. P. Lee, K. M. Lim, A survey of sentiment analysis: Approaches, datasets, and future research, Applied Sciences 13 (2023). URL: https://www.mdpi.com/2076-3417/13/7/4550. doi:10.3390/app13074550.

[9] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, 2013. arXiv:1310.4546.

[10] A. Severyn, A. Moschitti, Twitter sentiment analysis with deep convolutional neural networks, in: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, 2015, pp. 959–962. doi:10.1145/2766462.2767830.

[11] R. K. Behera, M. Jena, S. K. Rath, S. Misra, Co-lstm: Convolutional lstm model for sentiment analysis in social big data, Information Processing & Management 58 (2021) 102435. URL: https://www.sciencedirect.com/science/article/pii/S0306457320309286. doi:https://doi.org/10.1016/j.ipm.2020.102435.

[12] G.-D. Pilar, S.-B. Isabel, P.-M. Diego, G. Ávila José Luis, A novel flexible feature extraction algorithm for spanish tweet sentiment analysis based on the context of words, Expert Systems

with Applications 212 (2023) 118817. URL: https://www.sciencedirect.com/science/article/pii/S0957417422018358. doi:https://doi.org/10.1016/j.eswa.2022.118817.

[13] D. Elreedy, A. F. Atiya, A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance, Information Sciences 505 (2019) 32–64. URL: https://www.sciencedirect.com/science/article/pii/S0020025519306838. doi:https://doi.org/10.1016/j.ins.2019.07.070.

[14] A. S. Hussein, T. Li, D. M. Abd Ali, K. Bashir, C. W. Yohannese, A modified adaptive synthetic sampling method for learning imbalanced datasets, in: Developments of Artificial Intelligence Technologies in Computation and Robotics: Proceedings of the 14th International FLINS Conference (FLINS 2020), World Scientific, 2020, pp. 76–83.

[15] M. Aloraini, A. Khan, S. Aladhadh, S. Habib, M. F. Alsharekh, M. Islam, Combining the transformer and convolution for effective brain tumor classification using mri images, Applied Sciences 13 (2023). URL: https://www.mdpi.com/2076-3417/13/6/3680. doi:10.3390/app13063680.

[16] B. Jang, M. Kim, G. Harerimana, S.-u. Kang, J. W. Kim, Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism, Applied Sciences 10 (2020). URL: https://www.mdpi.com/2076-3417/10/17/5841. doi:10.3390/app10175841.

[17] G. Bel-Enguix, H. Gómez-Adorno, G. Sierra, J. Vásquez, S.-T. Andersen, S. Ojeda-Trueba, Overview of HOMO-MEX at Iberlef 2023: Hate speech detection in Online Messages directed tOwards the MEXican spanish speaking LGBTQ+ population, Procesamiento del lenguaje natural 71 (2023).

[18] D. M. Eler, D. Grosa, I. Pola, R. Garcia, R. Correia, J. Teixeira, Analysis of document pre-processing effects in text and opinion mining, Information 9 (2018). URL: https://www.mdpi.com/2078-2489/9/4/100. doi:10.3390/info9040100.

[19] D. Huerta-Velasco, H. Calvo, Verbal aggression detection in mexican tweets, Computacion y Sistemas 26 (2022) 261–269. doi:10.13053/CyS-26-1-4169, publisher Copyright: © 2022 Instituto Politecnico Nacional. All rights reserved.

[20] R. Silva Barbon, A. T. Akabane, Towards transfer learning techniques—bert, distilbert, bertimbau, and distilbertimbau for automatic text classification from different languages: A case study, Sensors 22 (2022) 8184.