

English Benchmark Hate Speech and Offensive Content Detection Based on Layer-by-Layer Classification

Kongqiang Wang^{1*}, Qingli Tan², and Peng Zhang¹

¹School of Information Science and Engineering, Yunnan University, 650500, Kunming, Yunnan, China

²College of Ecology and Environment, Yunnan University, 650500, Kunming, Yunnan, China

*Corresponding author: wangkongqiang60@gmail.com

Abstract

The ability to understand English text snippets content is an essential component of human-like artificial intelligence, as English text snippets content greatly influence human cognition, decision making, and social interactions. In addition to intention recognition in diffusion of harmful content on online web, the task of identifying the potential hate and offensive categories behind an individual's text state in English text snippets is of great importance in many application scenarios. The main content of our research is content classification and labeling from English text snippets of hate and offensive language in various social media comments content, which aims at assign a content classification label to English text snippets taken from social media content. Each snippet contains 1–3 sentences and corresponds to one of the predefined categories based on the English text comments. Our main task is Subtask 1 divided into two hierarchical sub-tasks on the English HASOC2021 dataset. Subtask 1A: Identifying hate, offensive and profane content from the post; and Subtask 1B: Discrimination between hate, profane and offensive posts; We used various machine learning (ML) and deep learning (DL) techniques and their results were compared. Subsequently, the pre-trained language model *DistilBERT* was used to identify hate, offensive and profane content in the posts. This is done in order to understand which prediction model framework is the most effective when predicting hate and offensive categories. We obtained the best prediction model through the layer-by-layer classification method in the Macro F1 scores of *Subtask 1A* and *Subtask 1B* were 76.00% and 58.80% respectively. The project code is available from <https://github.com/WangKongQiang/DHOW>.

Index Terms— Text Content Analysis, Text Classification, Machine Learning (ML), LSTM, DistilBERT, TF-IDF.

1 Introduction

Understanding English text snippets content is crucial to achieve human-like artificial intelligence, as English social media text snippets are intrinsic to humans life and significantly influence our cognition, decision-making, and social

interactions [1]. Various social media comments are an important form of human communication and contains a large amount of information. Furthermore, given that comments in their natural form are textual modality, many studies [2] have explored textual modality intention recognition in English text snippets content using language modalities.

There are increasing harmful comments in the news media, social media, and other web sources [3]. The harmful content on the internet should be classified into different categories based on their language characteristics. Much recent research [4] has focused on fake news and hate speech. Fake news is further classified as misinformation, disinformation, and malformation. In recent years, much research [5] has been done for fake news detection and debunking of fake news. In the last two decades, there has been a tremendous increase in the spread of misinformation, which is also reflected by the number of fact-checking websites [6]. Fact-checking websites can help investigate claims and assist citizens in determining whether the information used in an article is true. More than 100 fact-checking websites are working in 40+ languages across 100+ countries [7]. The problem of fake news is multilingual, with multiple platforms like X (previously Twitter), Facebook, YouTube, and encrypted platforms like WhatsApp. However, many of these fake news contain elements of hatred and offense. Hate Speech can be defined as text that expresses prejudice against a person from a group or a particular group, primarily based on race, religion, or sexual orientation. The message should be about a specific group based on race, religion, gender, class, ethnicity, sexual orientation, nationality, physical appearance, disability, or disease [8]. The research on identifying hate and offensive content can also be transferred and applied to the study of identifying fake news.

Hate speech can ultimately also lead to violent hate crimes [9]. Consequently, studies of detection and moderation hate and offending content are necessary to maintain a rational discourse that allows an exchange of arguments. Reduced efforts in content moderation can lead to the proliferation of hate speech, as the case of Twitter has shown recently [10]. One of the most effective strategies for tackling this problem is to use computational methods to detect different harmful content, and another is to identify the topic of hate speech. This topic has attracted significant attention in recent years as evidenced in recent publications, e.g. the research by Li et al. [11] and

workshops held in conjunction with *KDD* [12] and *Mediaeval* [13]. Most of the research has adopted a binary view of the classification problem (*false* or *true*), but for partially false news, different patterns of circulation have been observed and false news propagates faster than partially false news on social media [14]. This also occurs in the study of hate and offensive content. Completely hateful and offensive content spreads faster than partial content that supports hate and offense. We can also identify the content that is hateful or offensive to help make a decision on whether this news is a fake news story. This is done because many of these fake news stories contain elements of hatred and offense.

Previously, many approaches have been developed to improve the identification of hate speech in some shared tasks [15]. To address these challenges, this study draws upon principles from information theory and causal reasoning to construct a novel computational model tailored to social media comment texts. By incorporating both statistical properties of language and causal relationships among text embeddings, the proposed model aims to enhance the interpretability and analytical performance of automated systems. Ultimately, this work seeks to lay a solid theoretical and methodological foundation for the automated analysis of hate and offensive content in social media, contributing to more reliable and scalable evaluation of English social media text classification performance.

We proposing numerous well-designed pipeline systems. Moreover, we applied advanced pre-trained language model, deep learning (DL) model and machine learning (ML) models for hate, offensive, and profane text analysis and achieved promising results. These prediction models include *distilbert/distilbert-base-uncased*, *LSTM*, *Random Forest*, *SVM*, *Logistic Regression*, and *K-Nearest Neighbor (KNN)*.

2 Related Works

2.1 The Evolution of Deep Learning (DL) in Text Classification

Text classification, as a core task in natural language processing (NLP), has undergone a significant transformation from traditional machine learning (ML) to deep learning (DL). Early studies mainly relied on Bag-of-Words models and TF-IDF feature engineering methods [16], which performed well in short text classification but had difficulty capturing complex semantic relations. The introduction of convolutional neural networks (CNNs) has brought new breakthroughs to text classification. Yoon Kim et al. [17] proved that multi-scale convolutional kernels can effectively extract local semantic features. Subsequently, recurrent neural networks (RNN) especially long short-term memory networks (LSTM), have demonstrated advantages in processing long sequential texts [18]. The emergence of the Transformer architecture has completely transformed the technical landscape of text classification. BERT [19] has achieved significant performance im-

provements on multiple text classification tasks through bidirectional encoder and mask language model pre-training. Subsequent improved models such as RoBERTa [20] and DeBERTa [21] have further promoted the development in this field. However, these general pre-trained models still have domain adaptability issues in professional domain text processing, especially when dealing with English social media text with special language features.

2.2 Research on Domain-specific Text Classification

Text classification in professional fields is confronted with challenges such as terminology specialization, context complexity, and scarcity of labeled data. In the field of financial text analysis, Dogu Araci et al. [22] proposed the FinBERT model, which is specifically pre-trained for financial documents and performs well in financial sentiment analysis and document classification tasks. In the field of medical text classification, BioBERT [23] and ClinicalBERT [24] have effectively improved the classification performance of medical literature and clinical records through continuous pre-training on biomedical corpora. In the research of legal text classification, Ilias Chalkidis et al. [25] developed LegalBERT, which specifically handles the language features of legal documents. The success of these domain-specific models validates the importance of domain adaptation for professional texts. However, as an emerging professional field, the language features and classification requirements of social media comments laden with hatred and offensive language have not been fully studied.

3 Task and Dataset

3.1 Task Description

Social media today is a hotbed of growing hate speech. This has emerged as a critical challenge for governments globally. People are spending a considerable amount of time on social media like Facebook, Twitter, and Instagram. Studies [8] suggest that most of the online content generated on these platforms contain abusive language. There is a need to develop adequate response mechanisms in order to find a balance between freedom of expression on one side, the ability to live without oppressive remarks on the other side and a requirement for a robust technology to identify problematic content automatically.

HASOC [8] provides a forum for developing and testing text classification systems for various languages. It organized a shared task for FIRE 2021. The task is aimed at identifying hateful and offensive language in social media posts. The task was organized for two languages namely English and Hindi, but we conducted our investigation only on the English language dataset. Our team mainly participated in Subtask 1A: Identifying hate, offensive and profane content from the post and Subtask 1B : Discrimination between hate, profane and offensive posts. We achieved significant development results by using pre-trained language model technology.

There were two Subtasks given. Subtask 1: Identification and discrimination of hate, offensive and profane content from the post. Subtask 2: Identification of conversational hate-speech in code-mixed languages (ICHCL). We chose to carry out Subtask 1 from these. The problem statement we chose consists of hierarchical two-subtasks:

Subtask 1A: Identification of hate and offensive language, posts are classified as follows:

- **HOF**: Hate and Offensive. This post contains hate, offensive, and profane content.
- **NOT**: Non Hate and Offensive. This post does not contain any hate speech, profane, offensive content.

Subtask 1B: Categorization of hate and offensive, posts are further categorized as follows:

- **HATE**: Hate Speech. Posts under this class contain hate speech content.
- **OFFN**: Offensive Speech. Posts under this class contain offensive content.
- **PRFN**: Profane Speech. These posts contain profane words.

Various machine learning (ML) and deep learning (DL) models have been surveyed to check which produces more accuracy in detecting the various hate and offensive comments. A pre-trained language model DistilBERT is also used and it also inspired us to propose our innovate prediction model with the help of DistilBERT and other profanity detection libraries *profanityfilter* and *better-profanity* for classifying the *HOF* comments detected from Subtask 1A, as *HATE*, *OFFN* or *PRFN* in Subtask 1B.

3.2 Task Dataset

The datasets were annotated in a hierarchical fashion as shown below, see Table 1. Class distribution of the English dataset across Subtasks 1A and 1B.

Table 1: Details of English social media comments dataset are provided.

Task Details	Posts in Train Data	Posts in Test Data
SUBTASK 1A	<i>Total = 3843</i>	
Hate and Offensive posts (HOF)	2501	1281
Non Hate and Offensive posts (NOT)	1342	
SUBTASK 1B	<i>Total = 3843</i>	
Hate posts (HATE)	683	
Offensive posts (OFFN)	622	1281
Profane posts (PRFN)	1196	
NONE	1342	

In the training data for Subtask 1A, 65% of the data are hate and offensive comments (*HOF*) and about 35% are non hate and offensive comments (*NOT*). For Subtask 1B, the 65% hate and offensive comments (*HOF*) were classified as *HATE* (18% of the overall comments), *OFFN* (16% of the overall



Figure 1: This is an illustrative example for our task objective. The method we proposed is mainly aimed at eliminating the hateful and offensive content found in social media comments.

comments) and *PRFN* (31% of the overall comments). The training data will consist of English text snippets from different social media platforms, which contains target content and context content, an illustrative example of the task objective is shown in Figure 1.

4 Data analysis

4.1 Word Cloud

The following representation images are the word cloud for various inappropriate types of social media comments given in the HASOC2021 English dataset. It is a graphical representation for word frequency of different words used in each category of the comments. Figure 2 for word cloud of *HATE* comments in different social media platforms, Figure 3 for word cloud of *OFFN* comments in different social media platforms, Figure 4 for word cloud of *PRFN* comments and Figure 5 for word cloud of *NONE* comments.

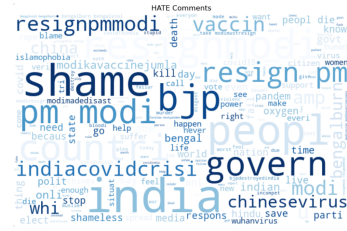


Figure 2: Word cloud for *HATE* comments.

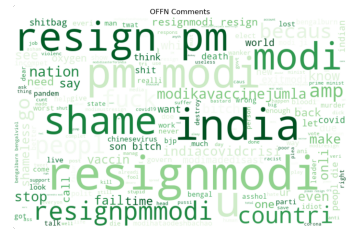


Figure 3: Word cloud for *OFFN* comments.

In the first step (text collection), the social media comments to be processed and collected. The second step (text pre-processing) involves the standardization of unstructured texts to increase the performance of natural language processing (NLP). The text collected contains many elements that are difficult to analyze, such as tags, references, abbreviations. Most are expressed as if speaking in a care-free manner, in terms of the vocabulary or the structural order of the sentence. Therefore, after text processing include change the uppercase to lowercase, delete special characters, tags, @'s, and remove stopwords, pre-processing is performed according to the requirement and stemming is also done. Finally, in the machine learning (ML) modeling stage, a supervised learning model is established, and training and prediction are performed using vectorized number type data using TF-IDF. In this study, we use the LSTM, DistilBERT and other basic machine learning (ML) models for training and prediction, These methods we proposed is mainly for the detection of hate speech, offensive language and profanity.

Pre-processing methods we used are shown in Figure 11. This picture shows a flowchart of text processing and machine learning (ML) model training. The flowchart consists of six steps from top to bottom in sequence. The first step is to remove unnecessary characters, including special characters, tags (such as hashtags in Twitter), and web addresses. The second step is to perform word segmentation on the tweets. We need to divide the comment text into individual words or tags. The third step is to remove stop words, which are usually common words that do not carry useful information in text analysis, such as "de", "yes", etc. The fourth step is to encode the categorical data using One Hot Encoder, which is a method of converting categorical variables into numerical forms that can be better processed by machine learning (ML) algorithms. The fifth step is to perform feature extraction and text encoding. This step involves extracting useful information from the text and converting it into a format that the model can handle. The final step is to train and predict using a machine learning (ML) model, which is the ultimate goal of the entire process. We need to classify or predict new data through the trained prediction model. The entire flowchart is connected by arrows, clearly showing the sequence and process of each step.

5.2 Feature Extraction

5.2.1 Term Frequency-Inverse Document Frequency (TF-IDF)

Initially, Bag of Words (BoW) [30] was used for training. However, it was quite inefficient in training the models owing to the fact that BoW doesn't consider term ordering and the rareness of the term. Hence, we used Term Frequency-Inverse Document Frequency (TF-IDF) [31] to overcome some of these drawbacks. The TF-IDF model contains information on the more important words and the less important ones as well. Thus performed well on the machine learning (ML) models.

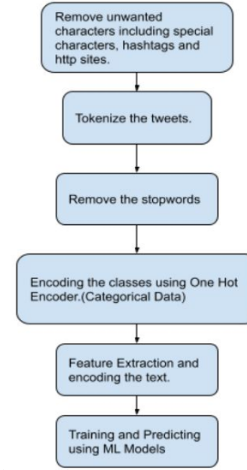


Figure 11: Overall data flow and the experiment methodology of machine learning (ML).

Table 2: The main machine learning (ML) method of our experiment.

Method	Full name	Comments
NB	GaussianNB	sklearn.naive_bayes
LR	LogisticRegression	sklearn.linear_model
KNN	KNeighborsClassifier	sklearn.neighbors
SVC	LinearSVC	sklearn.svm
Random Forest	RandomForestClassifier	sklearn.ensemble

5.2.2 BERT Encoding

The ktrain¹ library is used which contains the Transformers API, that allows the use of any Hugging Face transformers model and we used DistilBERT in our experiment. The pre-process train and test datasets were used for the pre-trained language model *distilbert/distilbert-base-uncased* before model training. Word embeddings are generated from the transformer model which is used for encoding. ktrain is a lightweight wrapper for the deep learning (DL) library TensorFlow Keras to help build, train, and deploy neural networks and other machine learning (ML) models to make deep learning and AI more accessible and easier to apply.

5.3 Classifiers

The following is a brief introduction to our machine learning (ML) methods. Most of these machine learning (ML) methods can be obtained in the scikit-learn package. They can classify more accurately through the detailed processing of text features, See Table 2. The configuration of the deep learning (DL) methods is shown in Table 3.

5.3.1 Machine Learning (ML)

For our experiment we used many different machine learning (ML) approaches. First, a frequency dictionary was built, with which we train the model and obtain the theta parameter for the sigmoid function that would better represent the comment

¹ <https://pypi.org/project/ktrain/>

Table 3: The main deep learning (DL) method of our experiment.

Layer (type)	Output Shape	Param
embedding (Embedding)	(None, 2500, 256)	2113280
lstm (LSTM)	(None, 64)	82176
dense (Dense)	(None, 2)	130

data. The accuracy for this prediction model came to 66% in Subtask 1A. To improve the accuracy we went ahead with TF-IDF vectorizer, which was used to train a Logistic Regression model. The model was imported from the scikit-learn python package. This improved the accuracy of the model to about 80% in Subtask 1A. For the next two models, we used a cross-validation approach with ten folds to train different machine learning (ML) models and saved the model with the best accuracy score. This validation feature was imported from the scikit-learn model-selection package. For the second approach, we went ahead with an SVM model. Given the task was a classification, SVM seemed to be an easier and efficient way to classify the tweet sentiments. The best model that was saved and it had an accuracy of about 80% in Subtask 1A [32].

To enable classification beyond binary scope, we went ahead to apply the K0.5 Nearest Neighbor Classification technique. We used the Euclidean Distance as the criterion to determine the clusters that would be formed. This was used specifically for Subtask 1B where there were four labels for classification namely *PRFN*, *OFFN*, *HATE*, *NONE*. The validation accuracy of this model came to about 75% for Subtask 1A and 57% for Subtask 1B. The last machine learning (ML) model, we went ahead with Random Forest classifier with entropy criterion. Random Forest tends to behave better in the case of noisy data as well. This pushed the accuracy of our prediction model to about 80% with Subtask 1A. Finally, we combined the predictions of all models together and took the majority of the prediction mode as the final prediction for Subtask 1A and Subtask 1B. See Table 4 and Table 5 for validation accuracy results for using machine learning (ML) models in Subtask 1A and Subtask 1B.

Table 4: Validation dataset experiment results for machine learning (ML) models in Subtask 1A.

Model	Accuracy	Macro-F1	Macro-Precision	Macro-Recall
Logistic Regression	80%	76%	80%	73%
SVC	80%	76%	76%	77%
KNN	75%	70%	72%	68%
Random Forest	80%	76%	76%	77%
Overall	78%	76%	77%	75%

Table 5: Validation dataset experiment results for machine learning (ML) models in Subtask 1B.

Model	Accuracy	Macro-F1	Macro-Precision	Macro-Recall
Logistic Regression	59%	52%	51%	54%
SVC	60%	52%	52%	53%
KNN	57%	47%	49%	46%
Random Forest	58%	50%	50%	50%
Overall	58%	50%	50%	51%

5.3.2 LSTM

Long Short-Term Memory (LSTM) [18] is a specific recurrent neural network (RNN) architecture that was designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs [33]. We used LSTM based neural network classifiers for both Subtask 1A and Subtask 1B. We used word tokens embedding layer 256 dimensions, input length 2500 words as the inputs to the LSTM 64 units layer with dropout of 20% and recurrent dropout of 20%, softmax layer for prediction in the keras toolkit. In this pipeline, we used binary-crossentropy as loss function, the Adam optimizer to optimize the parameters. See Table 6 for validation dataset experiment results for using LSTM models in Subtask 1A and Subtask 1B.

Table 6: Validation dataset experiment results for LSTM model.

Task	Accuracy	Macro-F1	Macro-Precision	Macro-Recall
Subtask 1A	78%	74%	76%	73%
Subtask 1B	60%	55%	55%	56%

5.3.3 DistilBERT

DistilBERT runs 60% faster while preserving over 95% of BERT's performance. It also uses less parameters than BERT [34]. It outperformed BERT and has now cemented itself as the model to beat for not only text classification, but also advanced NLP tasks [35]. Implementation of DistilBERT pipeline is much easier than using any other pre-trained learning and thus helpful in performing transfer learning. Hence, we chose to use DistilBERT² in our experiment. It contains 6 layers, 768 dimensions and 12 heads with 66 million parameters and is pretrained on BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia. For the English social media comments dataset, we used a pre-trained language model *distilbert/distilbert-base-uncased* for both Subtask 1A and Subtask 1B. The model was trained with train dataset, validated with the validation dataset and finally tested on the test set given. For training the max-len of the input sequence was chosen to be 500. If you set the max-length very high, you might face memory shortage problems during execution. On testing the prediction model on validation dataset (15% split from training dataset) the following results were inferred in Table 7.

Table 7: Validation dataset experiment results for DistilBERT model.

Task	Accuracy	Macro-F1	Macro-Precision	Macro-Recall
Subtask 1A	78%	73%	75%	72%
Subtask 1B	61%	53%	55%	52%

²https://huggingface.co/transformers/model_doc/distilbert.html

5.3.4 Proposed Prediction Model for Subtask 1B

This proposed prediction model uses the DistilBERT model trained for Subtask 1A. The motive behind using this model is its effectiveness on the validation dataset and the model's reliability [36]. The *HOF* comments are to be further classified as *PRFN*, *OFFN* or *HATE* comments. Proposed prediction model block diagram, see Figure 12.

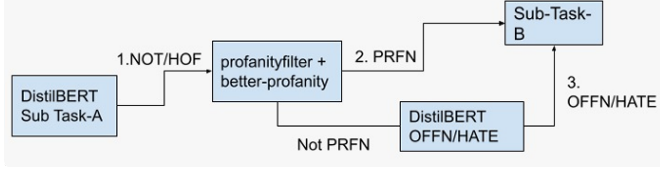


Figure 12: Proposed prediction model block diagram. The layer-by-layer classification method we proposed consists of three small steps.

There are three separate small modules combined to form the proposed prediction model. The three small modules are as follows:

1. The DistilBERT model trained for Subtask 1A identifies if a comment is of hate or not. This is essentially a binary classification (*HOF/NOT*).
2. Two python libraries *profanityfilter*³ and *better-profanity*⁴ are together used to classify the hate comments from the above model into *PRFN* or other hate and offensive comments.
3. Finally, another DistilBERT model is trained on the *OFFN* and *HATE* comments from the training sample dataset, is used to classify the remaining comments respectively.

The flow of the proposed prediction model is as follows: The test data is initially passed to DistilBERT model above, which is a binary classification model that returns if the given text is hate or not [37]. The non hate comments predicted by this model are labelled as *NONE*. And the hate comments are to be further classified as *HATE*, *PRFN* and *OFFN*. Now the hate comments in the test data as filtered by the first DistilBERT model as a part of the proposed model alone is passed to two python libraries *profanityfilter* and *better-profanity* that classify the hate comments into *PRFN* hate comments and other hate comments (*HATE* and *OFFN*) to be further classified. Lastly, the other hate comments in the test dataset is passed to another DistilBERT model trained to classify a text as either *HATE* or *OFFN*. The second DistilBERT model also trained using the training sample dataset provided, which will further classify the text as *OFFN* and *HATE* and ultimately all the text in the test dataset are now classified as *HATE*, *OFFN*, *PRFN* or *NONE*.

The layer-by-layer classification method [38] we proposed consists of three small steps. They have achieved good results

in the performance indicators at each stage as shown in Table 8. The best effectiveness was achieved in Subtask 1B as shown in Table 9.

Table 8: Validation dataset experiment results for each small module used in our proposed prediction model.

Module	Accuracy	Macro-F1	Macro-Precision	Macro-Recall
DistilBERT for Subtask 1A	78%	73%	75%	72%
Profanity filter + better-profanity	92%	48%	50%	46%
DistilBERT for OFFN/HATE classification	75%	73%	80%	74%

Table 9: Overall validation dataset experiment results for our proposed prediction model.

Task	Accuracy	Macro-F1	Macro-Precision	Macro-Recall
Subtask 1A	78%	73%	75%	72%
Subtask 1B	63%	59%	61%	58%

6 Result

The machine learning (ML) model's performance improved using K-Folding technique with 10 splitting iterations in the cross-validation [39]. Even though the validation performance of the machine learning (ML) models were decently high, the pre-trained language model DistilBERT proved to be most accurate on the test data on submission with a 77.67% accuracy score for Subtask 1A and around 65% accuracy for Subtask 1B. Our proposed prediction model gave the next best prediction for Subtask 1B with approximately 60% accuracy. The machine learning (ML) models did not perform well when for the Subtask 1B, a possible reason for it could be that these models could not differentiate between profane, hateful and offensive posts properly [40]. Data augmentation, exploring other pre-trained language models such as XLNet [41], ERNIE [42], RoBERTa [43], and considering POS tags combined with n-grams to give an extra set of feature space could be the scope of improvement to solve this problem [44]. These proposed methods experiment in test dataset for Subtask 1A and Subtask 1B, see Figure 13 and Figure 14.

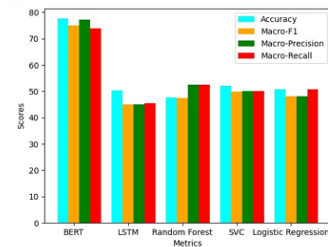


Figure 13: Test dataset experiment results for Subtask 1A.

³<https://pypi.org/project/profanity-filter/>

⁴<https://pypi.org/project/better-profanity/>

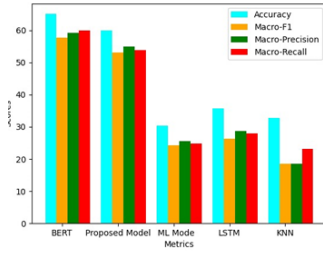


Figure 14: Test dataset experiment results for Subtask 1B.

6.1 Experiment Results Obtained for Subtask 1A

From the identification of hate and offensive content in the test dataset experiment results for different prediction models, it can be seen that the proposed prediction model achieves the best results with a Macro-F1 value of 76.00%, and also lays the foundation for the further implementation of Subtask 1B, see Table 10. The proposed prediction model based on pre-trained language model DistilBERT that achieves the best results in Subtask 1A. Its error analysis is based on the confusion matrix. Please refer to Figure 15.

Table 10: Test dataset experiment results for different machine learning (ML) models, deep learning (DL) model and Pre-trained language model for Subtask 1A.

Model	Accuracy	Macro-F1	Macro-Precision	Macro-Recall
Proposed Model	77.67%	76.00%	77.30%	74.91%
DistilBERT	77.63%	75.90%	77.28%	74.89%
LSTM	50.35%	45.25%	45.06%	45.48%
Random Forest	47.77%	52.70%	52.86%	52.61%
SVM	52.06%	50.00%	50.04%	50.05%
Logistic Regression	50.66%	48.15%	48.18%	48.14%
Overall	50.64%	48.13%	48.15%	48.12%

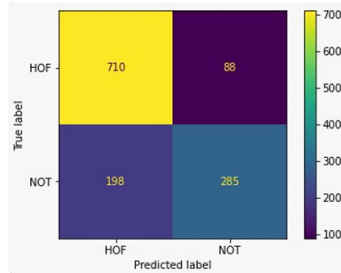


Figure 15: Subtask 1A: confusion matrix for proposed prediction model based on pre-trained language model DistilBERT.

6.2 Experiment Results Obtained for Subtask 1B

For Subtask 1B on the validation set, the prediction method we proposed slightly exceeds the Macro-F1 value of the DistilBERT Model which is 53% with a Macro-F1 value of 59%. However, on the test set, the prediction model we proposed

slightly exceeded the Macro-F1 value 54.43% of the DistilBERT model with a Macro-F1 value of 58.80%. For details, please refer to Table 11. The proposed prediction model based on pre-trained language model DistilBERT that achieves the best results in Subtask 1B. The prediction model we proposed achieved the first-place result. Its error analysis is based on the confusion matrix. Please refer to Figure 16. The pre-trained language model DistilBERT achieved the second-place result. Its error analysis is based on the confusion matrix. Please refer to Figure 17.

Table 11: Test dataset experiment results for different prediction models for Subtask 1B.

Model	Accuracy	Macro-F1	Macro-Precision	Macro-Recall
Proposed Model	65.10%	58.80%	59.90%	57.65%
DistilBERT	59.95%	54.43%	54.98%	53.91%
LSTM	35.75%	28.31%	28.66%	27.98%
KNN	32.86%	20.64%	18.58%	23.19%
Overall	30.44%	25.20%	25.57%	24.83%

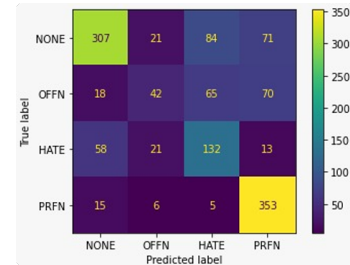


Figure 16: Subtask 1B: confusion matrix for proposed prediction model.

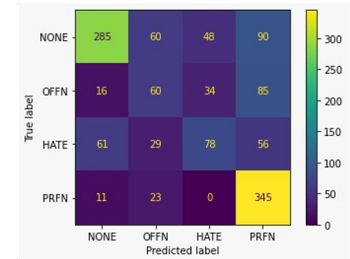


Figure 17: Subtask 1B: confusion matrix for DistilBERT pre-trained language model.

7 Conclusion

In this paper, several machine learning (ML) and deep learning (DL) approaches have been used for detecting hate speech and offensive language content and these models have been compared. Several attempts on various predictive techniques to increase the performance have been employed. Our proposed model achieved best results compared to its simplicity, better than the pretrained language model DistilBERT. We believe with proper feature extraction and data augmentation

techniques, we will be able to improve our proposed prediction model.

8 Acknowledgments

We are very grateful to the organizers and workers of HASOC 2021 for approving our dataset application and providing the training and test dataset.

References

- [1] Jie Tao, Xing Fang, "Toward multi-label sentiment analysis: a transfer learning based approach," *Journal of Big Data* (7), 1–26, 2020.
- [2] Uladzimir Sidarenka, "Potts: the potsdam twitter sentiment corpus," In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 1133–1141, 2016.
- [3] Hang Yan, Junqi Dai, Xipeng Qiu, Zheng Zhang, "A unified generative framework for aspect-based sentiment analysis," *arXiv preprint arXiv: 2106.04300*, 2021.
- [4] Rie Kubota Ando, Tong Zhang, Peter Bartlett, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research* (6), 1817–1853, 2005.
- [5] Xinyi Zhou, Reza Zafarani, "Fake news: A survey of research, detection methods, and opportunities," *arXiv preprint arXiv:1812.00315*, 2, 13, 2018.
- [6] Lucas Graves, Federica Cherubini, "The rise of fact-checking sites in Europe," 2016.
- [7] Gautam Kishore Shahi, Durgesh Nandini, "FakeCovid– A multilingual cross-domain fact check news dataset for COVID-19," *arXiv preprint arXiv:2006.11343*, 2020.
- [8] Sandip Modha, Thomas Mandl, Gautam Kishore Shahi, Hiren Madhu, Shrey Satapara, Tharindu Ranasinghe, Marcos Zampieri, "Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages and conversational hate speech," In *Proceedings of the 13th annual meeting of the forum for information retrieval evaluation*, pp. 1-3, 2021.
- [9] Karsten Müller, Carlo Schwarz, "From hashtag to hate crime: Twitter and antiminority sentiment," *American Economic Journal: Applied Economics* 15 (3): 270-312, 2023.
- [10] Daniel Hickey, Matheus Schmitz, Daniel Fessler, Paul E Smaldino, Goran Muric, Keith Burghardt, "Auditing Elon Musk's impact on hate speech and bots," In *Proceedings of the international AAAI conference on web and social media*, vol. 17, pp. 1133-1137, 2023.
- [11] Xian Li, Xin Luna Dong, Kenneth Lyons, Weiyi Meng, Divesh Srivastava, "Truth finding on the deep web: Is the problem solved?" *arXiv preprint arXiv:1503.00303*, 2015.
- [12] subho, "Fake News Detection Challenge KDD 2020," <https://kaggle.com/competitions/fakenewskdd2020>, Kaggle, 2020.
- [13] MediaEval, "FakeNews: Corona virus and 5G conspiracy," <https://multimediaeval.github.io/editions/2020/tasks/fakenews/>, 2020.
- [14] Gautam Kishore Shahi, Anne Dirkson, Tim A. Majchrzak, "An exploratory study of COVID-19 misinformation on Twitter," *Online social networks and media* 22: 100104, 2021.
- [15] Kanika Narang, Aida Mostafazadeh Davani, Lambert Mathias, Bertie Vidgen, Zeerak Talat, "Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)," In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, 2022.
- [16] Gerard Salton, Michael J McGill, "Introduction to modern information retrieval," McGraw-Hill, 1986.
- [17] Yoon Kim, "Convolutional neural networks for sentence classification," In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751, 2014.
- [18] Sepp Hochreiter, Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, 9(8):1735–1780, 1997.
- [19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
- [21] Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, "Deberta: Decoding-enhanced bert with disentangled attention," In *International Conference on Learning Representations*, 2021.
- [22] Dogu Araci, "Finbert: Financial sentiment analysis with pre-trained language models," *arXiv preprint arXiv:1908.10063*, 2019.
- [23] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, Jaewoo Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, 36(4):1234–1240, 2020.

- [24] Emily Alsentzer, John Murphy, William Boag, Weihung Weng, Di Jindi, Tristan Naumann, Matthew McDermott, "Publicly available clinical bert embeddings," In Proceedings of the 2nd Clinical Natural Language Processing Workshop, 72–78, 2019.
- [25] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, Ion Androutsopoulos, "Legal-bert: The muppets straight out of law school," In Findings of the Association for Computational Linguistics: EMNLP 2020, 2898–2904, 2020.
- [26] Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, "Multi-task deep neural networks for natural language understanding," In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 4487–4496, 2019.
- [27] Ronan Collobert, Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," In Proceedings of the 25th international conference on Machine learning, 160–167, 2008.
- [28] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy, "Hierarchical attention networks for document classification," In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, 1480–1489, 2016.
- [29] Sebastian Haunss, Jonas Kuhn, Sebastian Pado, Andre Blessing, Nico Blokker, Erenay Dayanik, Gabriella Lapesa, "Integrating manual and automatic annotation for the creation of discourse network data sets," Politics and Governance, volume 8(2), 2020.
- [30] Wisam A Qader, Musa M. Ameen, Bilal I. Ahmed, "An overview of bag of words; importance, implementation, applications, and challenges," In 2019 international engineering conference (IEC), pp. 200-204. IEEE, 2019.
- [31] Tim Loughran, Bill McDonald, "When is a liability not a liability? textual analysis, dictionaries, and 10-ks," The Journal of finance, 66(1):35–65, 2011.
- [32] Hamada A Nayel, H. L. Shashirekha, "DEEP at HASOC2019: A Machine Learning Framework for Hate Speech and Offensive Language Detection," In FIRE (working notes), pp. 336-343, 2019.
- [33] Alex Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," Physica d: Nonlinear phenomena 404: 132306, 2020.
- [34] Raphael Scheible, Johann Frei, Fabian Thomczyk, Henry He, Patric Tippmann, Jochen Knaus, Victor Jaravine, Frank Kramer, Martin Boeker, "Gottbert: a pure german language model," In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 21237-21250, 2024.
- [35] Erik Cambria, Bebo White, "Jumping NLP curves: A review of natural language processing research," IEEE Computational intelligence magazine 9, no. 2: 48-57, 2014.
- [36] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.
- [37] Julian Risch, Ralf Krestel, "Toxic comment detection in online discussions," In Deep learning-based approaches for sentiment analysis, pp. 85-109. Singapore: Springer Singapore, 2020.
- [38] Yuan-Pin Lin, Tzyy-Ping Jung, "Improving EEG-based emotion classification using conditional transfer learning," Frontiers in human neuroscience 11: 334, 2017.
- [39] Jakob Prange, Charlott Jakob, Patrick Göttfert, Raphael Huber, Pia Wenzel Neves, Annemarie Friedrich, "Overview of the SustainEval 2025 Shared Task: Identifying the topic and verifiability of sustainability report excerpts," In Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops, pp. 229-238, 2025.
- [40] Grigorios Tsoumakas, Ioannis Katakis, "Multilabel classification: An overview," International Journal of Data Warehousing and Mining (IJDWM), 3(3):1–13, 2007.
- [41] Yichun Yin, Yangqiu Song, Ming Zhang, "Document-level multi-aspect sentiment classification as machine comprehension," In Proceedings of the 2017 conference on empirical methods in natural language processing, 2044–2054, 2017.
- [42] Min-Ling Zhang, Zhi-Hua Zhou, "A review on multilabel learning algorithms," IEEE transactions on knowledge and data engineering, 26(8):1819-1837, 2013.
- [43] Telmo Pires, Eva Schlinger, Dan Garrette, "How multilingual is multilingual bert?" In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy. Association for Computational Linguistics, 4996–5001, 2019.
- [44] Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, Lidong Bing, "Sentiment analysis in the era of large language models: A reality check," In Findings of the Association for Computational Linguistics: NAACL 2024, pp. 3881-3906, 2024.