OFFENSIVE LANGUAGE AND HATE SPEECH DETECTION USING BERT MODEL

Fadila Shely Amalia[1] Yohanes Suyanto[2]
[1,2]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: **[1]fadilashelyamalia@mail.ugm.ac.id,** [2]yanto@ugm.ac.id

Abstrak

Deteksi ujaran kebencian dan bahasa ofensif merupakan isu krusial dalam analisis sentimen dan pemrosesan bahasa alami. Penelitian ini bertujuan untuk meningkatkan efektivitas deteksi ujaran kebencian dalam teks berbahasa Inggris dengan memanfaatkan BERT model. Selain itu, teknik preprocessing yang dimodifikasi juga dikembangkan guna meningkatkan nilai F1-score. Dataset yang digunakan dalam penelitian ini merupakan dataset public diambil Kaggle, yang berisi teks berbahasa Inggris dengan konten ujaran kebencian. Hasil evaluasi menunjukkan peningkatan signifikan dalam akurasi dan performa keseluruhan model dalam klasifikasi teks. Model BERT mencapai akurasi sebesar 89,11% pada data uji, dengan kemampuan prediksi yang tepat pada sekitar 85 dari 95 sampel. Analisis confusion matrix menunjukkan bahwa model sangat baik dalam mengklasifikasikan teks 'offensive' dengan akurasi sekitar 95%, namun menghadapi kesulitan dalam membedakan teks 'hate' dan 'offensive', serta terdapat kebingungan kecil antara teks 'neither' dan 'offensive'. Berdasarkan classification report, F1-score yang diperoleh adalah 0,43 untuk kelas 'hate', 0,94 untuk kelas 'offensive', dan 0,84 untuk kelas 'neither'. Weighted average F1-score mencapai 0,89, sementara macro average F1-score berada di angka 0,73. Hasil ini menunjukkan bahwa model BERT yang dilatih dengan pendekatan ini mampu memberikan performa yang solid dalam mendeteksi ujaran kebencian, meskipun masih terdapat ruang untuk perbaikan, khususnya dalam membedakan beberapa kelas tertentu.

Kata kunci*— Hate speech, Offensive, Deep Learning, BERT, Twitter*

Abstract

Hate speech detection is a crucial issue in sentiment analysis and natural language processing. This study aims to improve the effectiveness of hate speech detection in English text by utilizing the BERT model. Additionally, modified preprocessing techniques were developed to enhance the F1-score. The dataset used in this study was sourced from Kaggle, containing English text with hate speech content. Evaluation results show a significant improvement in the model's accuracy and overall performance in text classification tasks. The BERT model achieved an accuracy of 89.11% on the test data, correctly predicting about 85 out of 95 samples. Confusion matrix analysis revealed that the model performs very well in classifying offensive text with an accuracy of around 95%, but it struggles to distinguish between hate and offensive text, with slight confusion between neither and offensive texts. According to the classification report, the obtained F1-scores are 0.43 for the hate class, 0.94 for the offensive class, and 0.84 for the neither class. The weighted average F1-score is 0.89, while the macro average F1-score stands at 0.73. These results indicate that the BERT model trained with this approach is capable of delivering solid performance in detecting hate speech, although there is still room for improvement, particularly in distinguishing between certain classes.

Keywords— *Hate speech, Offensive, Deep Learning, BERT, Twitter*

1. INTRODUCTION

The widespread use of social media, which has become an integral part of daily life, has transformed how we communicate, share information, and interact. Along with the growing popularity of these platforms, new challenges have emerged, such as the spread of hate speech online. This makes hate speech detection a crucial focus in sentiment analysis and natural language processing, as it is essential to maintain a safe digital space free from harmful content [1].

Offensive language is a significant issue on social media platforms like Twitter. This affects the user experience and can have a negative impact on individuals, groups, and society at large. Offensive language has detrimental effects on individuals and society as a whole. Such content can cause stress, anxiety, and even psychological trauma to the targeted individuals. Additionally, hate speech can trigger social conflict, reinforce group divisions, and damage interpersonal relationships [2],[3], [4].

In their 2022 study, Roy, Bhawal, and Subalalitha focused on the rapid detection and removal of posts containing hate speech and offensive language from social media platforms. The research investigated and evaluated various machine learning and deep learning methodologies. The BERT model employed in this study demonstrated F1-scores of 0.76% and 0.88% on Malayalam and Tamil code-mixed datasets, respectively. [5]. However, after implementing the proposed ensemble framework, the results surpassed those of state-of-the-art models, achieving weighted F1-scores of 0.802 and 0.933 for Malayalam and Tamil code-mixed datasets, respectively. Additionally, several other studies have explored the topic of hate speech detection.[6], [7], [8], [9].

In their previous research, Roy, Bhawal, and Subalalitha (2022) the preprocessing steps were applied in the following order: normalizing contractions, converting all text to lowercase, replacing emojis with their textual descriptions, removing punctuation, eliminating extra spaces, and filtering out numbers and special characters, and limiting the dataset to only 4,000 data points. The data were labeled into two categories: OFF, which represents tweets containing offensive words, and NOT, which represents tweets without offensive content [5].

In this study, the researcher will use the same method as Roy et al. (2022) with the BERT Model, but with a different dataset and modifications to the preprocessing process. The preprocessing steps will include removing numbers and special characters, removing extra spaces, URL removal, letter normalization, contraction normalization, lemmatization, and handling typos. Thus, this research will produce new findings that strengthen the understanding that the method used can yield good or even better F1-scores with data and preprocessing enhancements to determine actions against hate speech on social media [5].

2. METHODS

This study will discuss the detection of hate speech on the social media platform Twitter. The research utilizes a Transformer Model, specifically the BERT Model, to address the challenges in detecting hate speech on social platforms. The study employs an English language dataset, specifically Twitter data containing tweets in English.

2. 1 Data descriptios

The dataset used in this study is derived from previous research by Raymond T. Mutanga (2020). It consists of 22,091 English-language data points sourced from the Twitter platform. Each data point in the dataset is a text from a tweet that may contain hate speech or offensive language. This dataset includes 916 hate speech tweets, 11,708 offensive tweets, and 2,432 neutral tweets or neither [9].

Table 1 The amount of data used in the research

Class	Data
Hate	916
Offensive	11708
Neither	2432
Total	15056

Table 1 presents the distribution of data across three classes used in the research. The Hate class comprises 916 data points, while the Offensive class contains the largest number, with 11,708 data points. The Neither class includes 2,432 data points. In total, the dataset consists of 15,056 data points. Through this classification we can identify and categorize tweets based on their content type whether it is offensive, hate, or neutral. The preprocessing techniques used in this research include:

- Removal of numbers and special characters,
- Removal of extra spaces,
- URL removal,
- Letter normalization,
- Contraction normalization,
- Lemmatization,
- Handling of typos.

The initial dataset consisted of five labels and after modification, the dataset will be reduced to three labels: hate, offensive, and neither. This dataset will serve as the foundation for detecting hate speech using ensemble techniques, with a focus on relevant content from the Twitter platform.

2. 2 Model

The BERT model is a type of transfer learning model that involves training a neural network for one task and then fine-tuning it for a new task. BERT utilizes the Transformer architecture, which includes an encoder to process input and a decoder to generate output. Unlike other unidirectional models that read text sequentially, BERT reads the entire sequence of words simultaneously, making it a bidirectional model. This allows BERT to understand contextual relationships between words in the text more accurately, as the encoder views the entire input without following a specific order, thereby capturing the context of surrounding words more effectively [3].

BERT introduced by Google is a model that despite its conceptual simplicity, demonstrates substantial empirical effectiveness. BERT has set new benchmarks by achieving state of the art performance in various classification tasks. Its deep bidirectionality is notable, as BERT simultaneously captures context from both left-to-right and right to left, enabling the learning of deep text representations. The BERT framework involves two sequential stages: pre-training, during which the model is trained on unlabeled data, and fine-tuning, where the model is adapted for specific NLP tasks. This model's capability to be trained on large datasets and subsequently applied to diverse language processing tasks eliminates the necessity of training from the ground up.. This allows for the utilization of knowledge already acquired by the model on new tasks, thereby saving time and resources required for model training. BERT is designed to use bidirectional representations, both left-to-right and right-to-left, simultaneously and integrates the MLM with NSP. As a result, BERT is considered one of the best methods for understanding text with complex contexts [10].

2. 2.1 BERT Base

This paper provides a thorough examination of BERT, a well-known deep learning language model, delving into how it works, its uses in different text analysis tasks, comparisons

with similar models, and its impact on natural language processing. Transforms language comprehension by capturing context from both directions within a sentence, similar to how the human brain processes information. This review aims to provide a thorough understanding of the BERT model and its diverse applications in different NLP tasks [11], [12]. Some common Transformer-based models include BERT, XLM-RoBERTa, and DistilBERT. m-BERT (Multilingual BERT) is used in this research and is based on the BERT architecture previously described. m-BERT has been trained on 104 monolingual corpora and is highly useful for multilingual text processing[13]. This research utilizes the base model described in Figure 1.

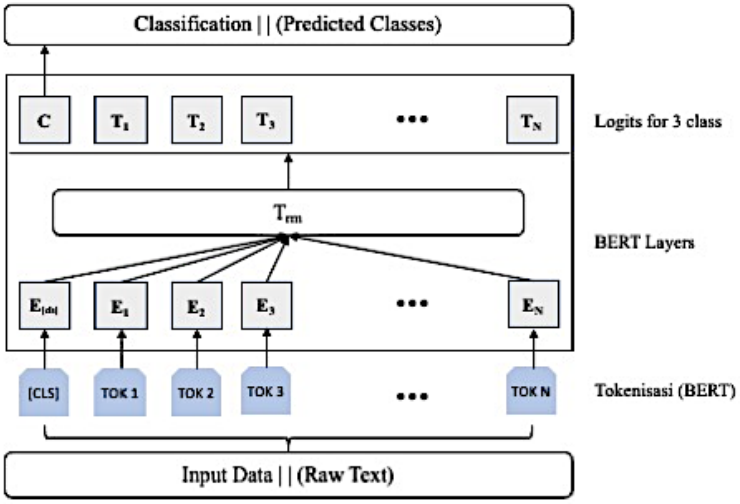

Figure 1 BERT model

In the first layer is the input layer. The process begins with raw text data from tweets to be classified. TOK1, TOK2, TOK3, TOK4, ... TOKN represent the tokenization process, which involves converting raw text into a numerical representation that the model can process. The text is transformed into token IDs using the BERT tokenizer. The [CLS] and [SEP] tokens are automatically added by the tokenizer but are not explicitly visible. The result of this process is a set of token IDs ready for further processing. E1, E2, E3, E4, ... EN represent the embedding process, where the token IDs are transformed into vector representations using the BERT embedding matrix. The BERT model used in this research includes an embedding layer that converts token IDs into semantically meaningful vector representations.

Model Architecture (TFBertForSequenceClassification) With the embeddings prepared, the data is fed into the BERT model architecture, which is adapted for classification tasks. TFBertForSequenceClassification is a BERT model with an additional layer for classification. This model consists of several transformer layers that process the token embeddings and produce more abstract representations of the input. The output from this layer is logits, which are raw scores indicating the likelihood that the input belongs to each of the defined classes. The output (T) is the result from the model, showing unnormalized scores for each class. Each score represents the model's confidence that the input belongs to a particular class. Classification (Predicted Classes) is the final result obtained. The class with the highest probability is the class predicted by the model for that input. This process provides the final output in the form of a class label, indicating the category or class of the tweet based on the analysis performed by the BERT model.

2. 2.2 DistilBERT

DistilBERT's performance has been evaluated on various benchmarks, such as the General Language Understanding Evaluation (GLUE) benchmark. It consistently shows comparable or improved results over the ELMo baseline and performs remarkably well compared to BERT, with significantly fewer parameters [14].

2. 2.3 Evaluation Method

In this study, I used several evaluation metrics, including accuracy, precision, recall, and F1-score to assess the performance of the classification model. These metrics are particularly useful for classification tasks with imbalanced classes or when certain class labels are more critical. Precision evaluates the proportion of correctly identified positive classifications, while Recall measures the proportion of actual positive cases that are correctly identified. The F1-score provides a balance between precision and recall by combining both metrics into a single measure.

3. RESULTS AND DISCUSSION

Training for BERT model was conducted for 8 epochs, with each epoch taking approximately 254-306 seconds about 4-5 minutes to complete. In the first epoch the training loss was around 0.4936 with an accuracy of about 82.89%, while the validation loss was about 0.3387 with an accuracy of around 87.58%. By the last epoch (epoch 8) the training loss was approximately 0.0568 with an accuracy of around 98.32%, while the validation loss was about 0.4944 with an accuracy of around 89.21%. Generally, the model's performance improved with the increasing number of epochs, indicated by a decrease in loss and an increase in accuracy for both training and validation data. However, there was a rise in validation loss, particularly after epoch 5, which may suggest overfitting. Using an early stopping callback, training was halted after the 8th epoch due to no significant improvement in validation performance after several epochs. This result indicates that the BERT model was well-trained and achieved high accuracy on the validation data, but attention should be given to potential overfitting after several epochs. Table 2 presents the results of training the BERT model.

Table 2 Results of BERT training

Epoch	Loss	Accuracy	Val_loss	Val_accuracy	Time per Epoch
1	0.4936	0.8289	0.3387	0.8758	306s
2	0.2916	0.8969	0.3205	0.8821	254s
3	0.2366	0.9150	0.2989	0.8911	254s
4	0.1891	0.9338	0.3148	0.8928	254s
5	0.1484	0.9496	0.3732	0.8871	254s
6	0.1150	0.9613	0.4400	0.8642	254s
7	0.0890	0.9700	0.4277	0.8835	254s
8	0.0568	0.9832	0.4944	0.8921	254s

Distil-BERT was trained for 7 epochs, with each epoch taking between 138 to 164 seconds about 2-3 minutes to complete, indicating a fairly consistent duration for each epoch. In the first epoch, the training loss was around 0.4365 with an accuracy of approximately 84.61%, while the validation loss was about 0.3347 with an accuracy of 88.35%. This result shows that the model began to learn and achieved a fairly good accuracy from the start. By the last epoch (epoch 7), the training loss decreased to 0.1454 with a significant increase in accuracy to 95.23%, indicating that the model learned the patterns in the training data very well. However, the validation loss increased to 0.4108, with accuracy slightly decreasing to 87.85%, suggesting potential overfitting, where the model became too tailored to the training data and performed worse on the validation data. Overall, the model's performance trend shows improved accuracy on the training data with decreasing loss, indicating better understanding of the training data. However, starting from epoch 5, there are signs of overfitting, as indicated by the increase in validation loss despite stable or slightly decreasing validation accuracy. The increase in validation loss after epoch 5 indicates potential overfitting, where the model becomes too specific to the training data and loses its ability to generalize well to unseen data. Therefore, it is important to

consider measures such as early stopping or regularization to prevent overfitting and enhance the model's generalization capability on new data.

Table 3 Results of BERT training

Epoch	Loss	Accuracy	Val_loss	Val_accuracy	Time per Epoch
1	0.4365	0.8461	0.3347	0.8835	164s
2	0.3068	0.8944	0.3225	0.8845	139s
3	0.2670	0.9043	0.3240	0.8805	138s
4	0.2317	0.9197	0.3403	0.8838	138s
5	0.2040	0.9296	0.3352	0.8861	138s
6	0.1708	0.9421	0.3694	0.8861	138s
7	0.1454	0.9523	0.4108	0.8785	139s

Table 4 presents a comparative evaluation of the BERT and DistilBERT models across various classes related to text classification tasks, specifically focusing on hate speech, offensive language, and neutral content, as well as the weighted average to provide an overview of the model's performance in the context of class imbalance.

Table 4 Evaluation results for the BERT and DistilBERT models

Model	*Class*	*Result*		
		Pecision	*Recall*	*F1-score*
BERT	*Hate*	0.47	0.39	0
	Offensive	0.92	0.95	0.94
	Neither	0.88	0.80	0.84
	Weighted Avg	0.89	0.89	**0.89**
distilBERT	*Hate*	0.48	0.34	0.40
	Offensive	0.92	0.94	0.93
	Neither	0.81	0.84	0.82
	Weighted Avg	0.88	0.88	0.88

For the base BERT model, the precision for the hate class is 0.47 and recall is 0.39, with an F1-score of 0.43. Although the precision is relatively higher compared to recall, the F1-score indicates that the model has a moderate performance in identifying the hate class. This performance suggests challenges in detecting the hate class, possibly due to data imbalance or difficulty in recognizing more complex hate patterns. For the offensive class, BERT demonstrates very good performance with precision of 0.92, recall of 0.95, and an F1-score of 0.94. This indicates that the model is highly effective in identifying the offensive class, likely due to a larger volume of data and more distinct patterns in this category. For the neither class, BERT achieves a precision of 0.88 and recall of 0.80, with an F1-score of 0.84. Although not as high as the offensive class, these results show that the model is capable of effectively identifying content that does not fall into the hate or offensive categories. Overall BERT's weighted average are 0.89, indicating strong overall performance, with notable strengths in detecting the offensive and neither classes.

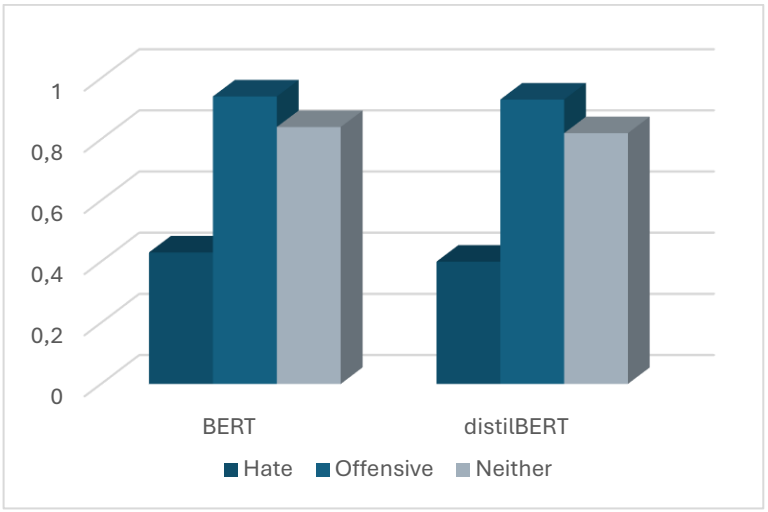

Figure 2 F1-score using different transformer models

In the application of DistilBERT, the hate class shows a slight increase in precision to 0.48 compared to BERT, but recall decreases to 0.34, with an F1-score of 0.40. The performance of DistilBERT on the hate class indicates that while precision is improved, the lower recall suggests that this model still faces challenges in effectively detecting the hate class. For the offensive class, DistilBERT achieves a precision of 0.92 and recall of 0.94, with an F1-score of 0.93. The performance on the offensive class is almost comparable to that of BERT, demonstrating that DistilBERT handles offensive classification well. The neither class shows a decrease in precision to 0.81 compared to BERT, but recall improves to 0.84, with an F1-score of 0.82. This indicates that although precision for the neither class is slightly lower, the increased recall suggests that the model is better at identifying the neither class overall. The weighted average precision, recall, and F1-score for DistilBERT are 0.88, 0.88, and 0.88, respectively. This indicates that although DistilBERT performs slightly worse than BERT overall, it still provides good performance with minimal differences in evaluation metrics.

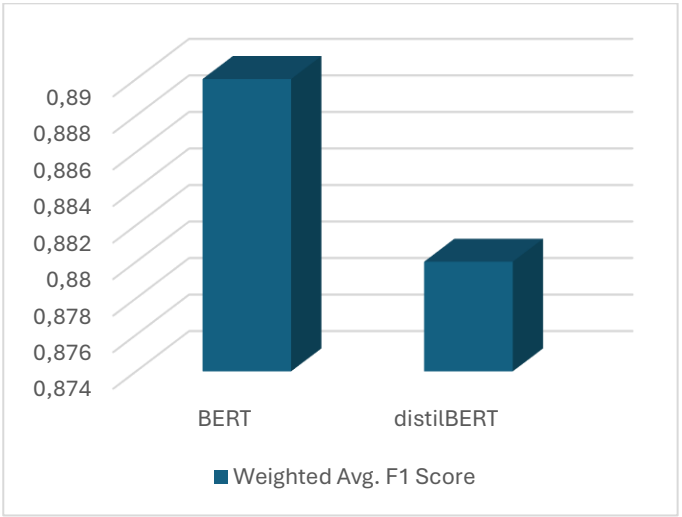

Figure 3 The F1-score obtained using different transformer models

In general, both BERT and DistilBERT demonstrate good performance in text classification with some differences across specific classes. BERT excels in detecting the offensive class, while DistilBERT shows competitive results but with variability in the hate and neither classes. The confusion matrix for the BERT model, as shown in Figure 2, illustrates

BERT's performance in classifying text into three categories: hate, offensive, and neither. This matrix provides insight into how well the model predicts each category compared to the actual labels.

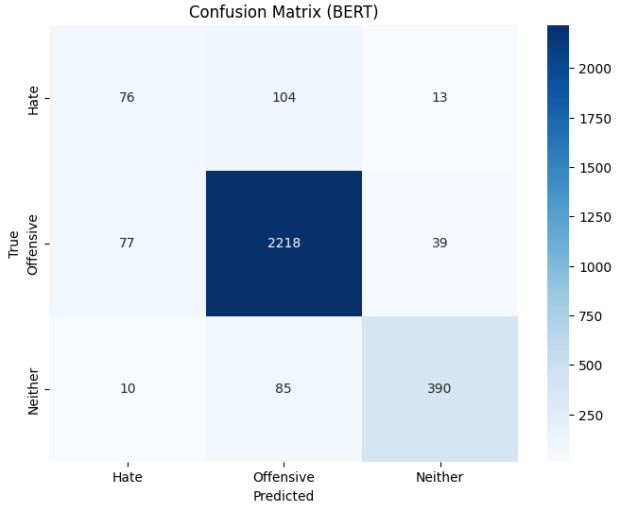

Figure 4 Confusion Matrix BERT Model

Class 0 (Hate): 76 samples that are truly categorized as hate are correctly predicted by the model as hate. 104 samples that are actually hate are incorrectly classified as offensive. 13 samples that are actually hate are incorrectly classified as neither. Class 1 (Offensive): 77 samples that are actually offensive are incorrectly classified as hate. 2,218 samples that are truly offensive are correctly predicted by the model as offensive. 39 samples that are actually offensive are incorrectly classified as neither. Class 2 (Neither): 10 samples that are actually neither are incorrectly classified as hate. 85 samples that are actually neither are incorrectly classified as offensive. 390 samples that are truly neither are correctly predicted by the model as neither.

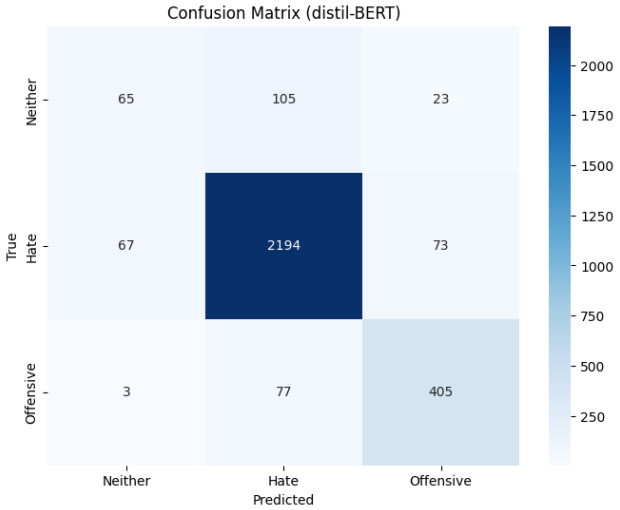

Figure 5 Confusion Matrix distilBERT Model

Based on the confusion matrix of the DistilBERT model, the performance of the model in classifying samples into three categories (hate, offensive, and neither) can be analyzed as follows: For the hate class (Class 0), the model correctly predicted 65 samples as hate. However, the model also made errors by classifying 105 samples that are actually hate as offensive, and 23 samples as neither. For the offensive class (Class 1), the model performed very well, correctly predicting

2,194 samples that truly belong to this category. Nevertheless, there were 67 samples that are actually offensive but incorrectly classified as hate, and 73 samples incorrectly classified as neither. For the neither class (Class 2), the model correctly classified 405 samples. However, it also made errors by classifying 3 samples as hate and 77 samples as offensive. Overall, the confusion matrix indicates that the DistilBERT model has good performance, particularly in detecting the offensive class, but still shows some weaknesses in distinguishing between the hate and neither classes.

4. CONCLUSIONS

This study successfully enhanced the effectiveness of hate speech detection in English text by implementing BERT model. The BERT model achieved an accuracy of 89.11% on the test data, indicating that the model was able to correctly predict approximately 85 out of 95 samples. Analysis of the confusion matrix shows that the model performs best in classifying offensive text with an accuracy of around 95%, but faces challenges in distinguishing between hate and offensive text, and also shows some confusion between neither and offensive text. From the classification report, the F1-scores obtained are 0.43 for the hate class, 0.94 for the offensive class, and 0.84 for the neither class. The weighted average F1-score is 0.89, while the macro average F1-score is 0.73. Overall, despite some weaknesses in distinguishing between certain categories, the BERT model provides satisfactory evaluation results and performs well for text classification tasks on the given dataset.

REFERENCES

[1] A. Matamoros-Fernández and J. Farkas, Racism, Hate Speech, and Social Media: A Systematic Review and Critique, *Television and New Media*, vol. 22, no. 2, pp. 205–224, Feb. 2021, doi: 10.1177/1527476420982230.

[2] S. E. Kapolri, T. Penanganan, U. Kebencian, M. Choirul, A. Dan, and M. Hafiz, Surat Edaran Kapolri Tentang Penanganan Ujaran Kebencian (Hate Speech) dalam Kerangka Hak Asasi Manusia.

[3] A. Mousa, I. Shahin, A. B. Nassif, and A. Elnagar, Detection of Arabic offensive language in social media using machine learning models, *Intelligent Systems with Applications*, vol. 22, Jun. 2024, doi: 10.1016/j.iswa.2024.200376.

[4] C. D. Putra and H.-C. Wang, Advanced BERT-CNN for Hate Speech Detection, *Procedia Comput Sci*, vol. 234, pp. 239–246, 2024, doi: 10.1016/j.procs.2024.02.170.

[5] P. K. Roy, S. Bhawal, and C. N. Subalalitha, Hate speech and offensive language detection in Dravidian languages using deep ensemble framework, *Comput Speech Lang*, vol. 75, Sep. 2022, doi: 10.1016/j.csl.2022.101386.

[6] J. A. Benítez-Andrades, Á. González-Jiménez, Á. López-Brea, J. Aveleira-Mata, J. M. Alija-Pérez, and M. T. García-Ordás, Detecting racism and xenophobia using deep learning models on Twitter data: CNN, LSTM and BERT, *PeerJ Comput Sci*, vol. 8, 2022, doi: 10.7717/PEERJ-CS.906.

[7] A. K. Das, A. Al Asif, A. Paul, and M. N. Hossain, Bangla hate speech detection on social media using attention-based recurrent neural network, *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 578–591, Jan. 2021, doi: 10.1515/jisys-2020-0060.

[8] A. Dewani, M. A. Memon, and S. Bhatti, Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for Roman Urdu data, *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00550-7.

[9] R. T. Mutanga, N. Naicker, and O. O. Olugbara, Detecting Hate Speech on Twitter Network Using Ensemble Machine Learning. [Online]. Available: www.ijacsa.thesai.org

[10] G. Z. Nabiilah, S. Y. Prasetyo, Z. N. Izdihar, and A. S. Girsang, BERT base model for toxic comment analysis on Indonesian social media, in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 714–721. doi: 10.1016/j.procs.2022.12.188.

[11] D. Yang, X. Wang, and R. Celebi, Expanding the Vocabulary of BERT for Knowledge Base Construction, 2023. [Online]. Available: http://ceur-ws.org

[12] A. A. Mosaed, H. Hindy, and M. Aref, BERT-Based Model for Reading Comprehension Question Answering, in *2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2023, pp. 52–57. doi: 10.1109/ICICIS58388.2023.10391167.

[13] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. [Online]. Available: https://github.com/tensorflow/tensor2tensor

[14] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.01108