Optimizing Hate Speech Detection in the Indonesian Language Using FastText and LSTM Algorithms

Febby Apri Wenando
Department of Information System
Universitas Andalas,
Padang, Indonesia
febby.apri@it.unand.ac.id
Department of Data Science
Universiti Malaysia Kelantan
Kelantan, Malaysia
j19e0033f@siswa.umk.edu.my

Nooraini Yusoff
Department of Data Science
Universiti Malaysia Kelantan
Kelantan, Malaysia
nooraini.y@umk.edu.my

Nurul Izrin
Department of Software Engineering
Universiti Teknikal Malaysia
Malaysia
izrin@utem.edu.my

2024 2nd International Symposium on Information Technology and Digital Innovation (ISITDI) | 979-8-3503-8924-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISITDI62380.2024.10796810

***Abstract*— The term hate speech encompasses communication that conveys hostility or prejudice toward individuals or groups based on specific attributes, including race, ethnicity, gender, or religion. Given the prevalence of such harmful content on major social media platforms like Twitter, it is imperative to prioritize ongoing research into automated detection methods. This study employs advanced artificial intelligence and machine learning techniques, involving pivotal stages such as data collection, preprocessing, oversampling, fastText word embedding, and classification using Long Short-Term Memory (LSTM) networks to achieve the objective. The amalgamation of fastText embeddings with LSTM classification yielded noteworthy results, with the model achieving 94% accuracy, 92% precision, 94% recall, and a 93% F1 score. These outcomes underscore the effectiveness of sophisticated machine learning methodologies in accurately discerning and addressing the proliferation of hate speech online, particularly in the context of Indonesian-language content.**

Keywords—FastText, LSTM, Adasyn, Imbalanced class, Indonesia Language

I. INTRODUCTION

The pervasive presence of social media in society has rendered it an indispensable platform for communication, information dissemination, and the articulation of opinions. Social media utilization's implications can manifest in positive and negative outcomes for individuals' lives. For instance, Twitter, a prominent social media platform, boasts a substantial user base in Indonesia. According to the Director-General of Resources for Post and Information Technology (SDPP) at the Ministry of Communication and Information, 19.5 million Indonesians are active Twitter users, positioning Indonesia as the fifth most active country on the platform [1].

One pertinent issue associated with social media, particularly Twitter, is the prevalence of hate speech[2]. Hate speech encompasses any form of communication expressing hatred towards individuals or groups [3]. The proliferation of user-generated web content, especially on social media platforms where individuals can freely express their opinions without stringent regulations, has resulted in a rise in incidents of hate speech. [3], [4], [5], [6]. Data from the Ministry of Communication and Information Technology indicates a 23% increase in reported hate speech incidents on social media platforms in Indonesia over the past year. This upsurge in hate speech poses significant risks to social harmony and individual well-being, rendering it a critical area of concern for policymakers, social media companies, and researchers alike.

Despite various efforts to tackle hate speech on social media, current models often encounter challenges in accurately identifying hate speech due to the nuanced and context-dependent nature of language. Conventional methods, such as keyword-based filtering, prove inadequate in capturing the subtleties and complexities of hate speech, resulting in high rates of false positives and negatives. Furthermore, the predominant focus of existing studies and models on English language datasets leaves a notable gap in research specifically addressing hate speech in the Indonesian language on Twitter [2], [6], [7], [8], [9].

In addressing this issue, an effective automated method for detecting hate speech involves the utilization of a machine learning (ML) framework [2]. Among advanced ML techniques, the Long Short-Term Memory (LSTM) algorithm has demonstrated significant promise. LSTM is well-suited for sequence prediction problems given its capacity to maintain long-term dependencies and address the vanishing gradient problem more effectively than traditional RNNs. This characteristic renders LSTM particularly advantageous for processing sequential data, such as text, wherein context and order are paramount for understanding meaning. Recent studies [10] have indicated that combining LSTM with FastText, a word embedding technique, yields impressive accuracy rates in text classification tasks. Although recent advancements in machine learning processing show promise in enhancing hate speech detection, there remains a paucity of models that integrate contextual and sequential information to improve detection accuracy. Specifically, the application of advanced machine learning techniques such as Long Short-Term Memory (LSTM) networks in conjunction with word embedding methods like FastText has received limited exploration in the context of Indonesian hate speech on Twitter.

The objective of this research is to create a strong model for identifying hate speech on Twitter using FastText word embeddings alongside the LSTM algorithm. Furthermore, the study aims to assess the performance of the proposed model, emphasizing the effectiveness of combining FastText and

LSTM to achieve high accuracy, precision, recall, and F1-score metrics in the context of hate speech detection.

II. RELATED WORK

The identification of hate speech on social media platforms has emerged as a major focus of research in natural language processing (NLP). Numerous studies have investigated different methods and techniques to enhance the precision and dependability of hate speech detection systems.

The range of text mining techniques applicable to the automated detection of hate speech is considerable, and comprehensive research is required to address the diverse aspects involved, including sentiment analysis, thematic classification and contextual understanding[11]. An effective approach to hate speech detection requires the analysis and extraction of significant information from extensive collections of user-generated content on social media platforms. Although the potential applications of text mining are extensive and the need for advanced techniques is pressing, the research output in this field remains relatively limited and highly relevant. This highlights the necessity for continued investigation and the development of sophisticated algorithms in order to enhance the online detection, and subsequent mitigation, of hate speech.

A. Detection of Hate Speech

The identification of hate speech involves the recognition and classification of text expressing hatred or bias towards individuals or groups based on specific characteristics such as race, ethnicity, gender, sexual orientation, nationality, religion, or other identifying factors. Previous studies have employed a range of machine learning (ML) and deep learning (DL) techniques for this purpose. For example, Davidson et al. assembled a dataset and utilized logistic regression and random forest classifiers to detect hate speech on Twitter, underscoring the challenge of discerning hate speech in the Indonesian language. [1], [12], [13].

B. Word Embeddings

Word embeddings have become a standard approach for representing text data in NLP tasks. Techniques like Word2Vec [14] and GloVe [15] are commonly used. FastText, which includes subword information to handle out-of-vocabulary words better, outperforms other embedding methods in various text classification tasks and hate speech detection [16], [17], [18], [19], [20].

C. Application of LSTM Networks

Recurrent neural networks, particularly Long Short-Term Memory (LSTM) networks, are adept at processing sequential data due to their ability to maintain long-term dependencies. LSTMs have found successful applications in various natural language processing (NLP) tasks, including language modeling, machine translation, and text classification. For instance, Syam et al. leveraged LSTM networks in combination with gradient-boosted decision trees to effectively detect hate speech on Twitter, resulting in high accuracy and F1-scores [10].

D. Combining FastText and LSTM

Recent research highlights the advantages of integrating word embeddings with LSTM networks for text classification tasks. FastText embeddings capture semantic relationships between words, while LSTM networks effectively model the sequential nature of text data. This combination was explored for sentiment analysis, achieving impressive results[21], [22], [23], [24].

The Indonesian language processing domain examines the use of FastText and LSTM for sentiment analysis. This research demonstrates the potential of these methods for the effective handling of Indonesian text [17]. Nevertheless, there has been no specific research conducted on the use of FastText and LSTM to detect hate speech in Indonesian contexts.

E. Recent Advances in Hate Speech Detection

There have been significant advances in hate speech detection methods in recent years. Kumar et al. [14] proposed a hybrid model that integrates FastText and Bidirectional Long Short-Term Memory (BiLSTM) networks for the detection of hate speech on the social media platform Twitter. This approach demonstrated a significant increase in classification accuracy compared to existing methods. Arango et al. [23] thoroughly evaluated various deep learning frameworks, including LSTM and FastText, for hate speech identification, underscoring the critical role of contextual understanding in enhancing model performance. Zhang et al.[25] introduced an efficient and lightweight model leveraging FastText embeddings and a CNN-LSTM architecture, which proved robust across multiple datasets [8]. Furthermore, Mandl et al. [26] investigated the challenges of multilingual hate speech detection using FastText and LSTM, addressing the complexities involved in cross-linguistic and cultural variations. Additionally, Wang et al.[27] developed an attention-based LSTM model augmented with FastText embeddings, aiming to improve contextual comprehension in hate speech detection.

FastText, developed by Facebook's AI Research (FAIA.I. team, provides an efficient method for word vectorization and text classification. In contrast to traditional word embedding methods like Word2Vec and GloVe, FastText incorporates subword information, enhancing its capability to handle out-of-vocabulary words. A comparative study was conducted to evaluate various word embeddings techniques in the context of text classification tasks, encompassing Word2Vec, GloVe, and FastText. The findings revealed that the FastText technique demonstrated superior performance compared to the other techniques, with an F-Measure accuracy of 97% [21]. Additionally, FastText has shown to be more effective than traditional TF-IDF methods, which do not capture semantic relationships between words and fail to handle out-of-vocabulary terms efficiently.

F. Scope and Relevance of Text Mining in Hate Speech Detection

The application of text mining methods for the automated identification of hate speech presents significant opportunities and warrants comprehensive exploration across various dimensions, including sentiment analysis, thematic classification, and contextual comprehension. The effective detection of hate speech involves the analysis and extraction of valuable insights from extensive user-generated content on social media platforms. Despite the broad relevance and pressing demand for advanced text mining techniques, the body of research in this domain remains somewhat limited

but exceedingly pertinent. This underscores the imperative for sustained inquiry and the advancement of sophisticated algorithms to bolster the online identification and mitigation of hate speech.

III. METHODOLOGY

A. Research Methodology Flowchart

In this study, several steps were taken to implement FastText for hate speech detection. These steps are shown in Fig. 1. The stages include data collection, pre-processing, oversampling, FastText word weighting, classification using LSTM, and evaluation.

This study carried out several stages of hate speech detection, as shown in Fig.1. as follows :

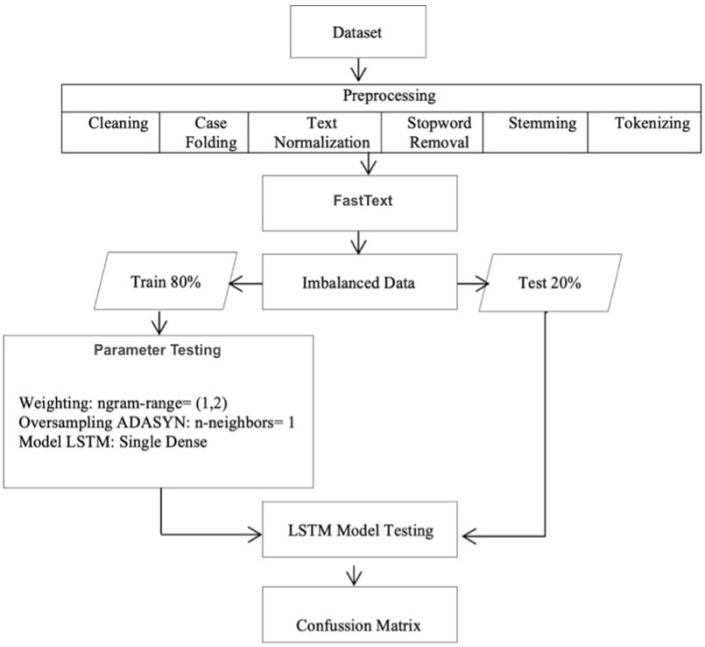


Fig.1. Flowchart

The diagram in Fig. 1 depicts the preprocessing stages, encompassing cleaning, case folding, normalization, stopword removal, stemming, and tokenization. Following preprocessing, data balancing is conducted through oversampling. Subsequently, word weighting is implemented using FastText, and the processed data is subjected to classification using the Long Short-Term Memory (LSTM) algorithm. The final phase entails evaluating the classification performance to assess the system's accuracy.

B. Data Collection

The data used is obtained from a dataset of Indonesian-language tweets taken from the research of Ibrahim & Budi[28], which have been classified into HS (Hate Speech) and NonHS (Non-Hate Speech) classes, totalling 13,169 tweets. This dataset is crucial as it provides a labelled collection of tweets, allowing supervised learning methods to be applied. The diversity and size of the dataset helps to train robust models that can generalise well to unseen data

C. Data Preprocessing

Preprocessing plays a critical role in the preparation of raw text data for analysis. It encompasses a series of sub-processes aimed at cleansing and normalizing the text, thereby addressing potential issues that may impact the accuracy of the model's results. The processes involved are as follows:

- Cleaning: Remove unnecessary characters such as numbers, special symbols, and punctuation marks that do not add to the meaning of the text.
- Case Folding: Convert all letters to lowercase to ensure consistency throughout the dataset.
- Text Normalization: Standardizing text by replacing abbreviations and slang words with their standard forms.
- Stopword Removal: The exclusion of common words that carry minimal semantic value, such as "and", "the", "is", etc., using the PySastrawi library.
- Stemming: This process involves reducing words to their base or root form to ensure consistent word.
- Tokenization: This process entails the segmentation of text into individual words or tokens

D. FastText Word Weighting

FastText builds word vectors by summing the vectors of their constituent n-grams, ensuring effective representation of words not present in the training corpus by their n-grams present in the corpus[20]. By incorporating subword information, FastText significantly improves upon traditional word embeddings, making it particularly advantageous for handling morphologically rich languages such as Indonesian.

In the FastText framework, the vector representation for a word (w) is computed as in (1):

$$\mathbf{v}(w) = \sum_{g \in G(w)} \mathbf{v}(g) \qquad (1)$$

Where ;

- $v(g)$ signifies the vector representation of n-gram g
- $g(w)$ denotes the set of n-grams generated from the word w

A pseudocode representation of the FastText word weighting process as follows in Table I:

TABLE I PSEUDOCODE OF THE FASTTEXT

Pseudocode for FastText word vector calculation

function compute_word_vector(word, ngrams, vector_dim):
  word_vector = zeros(vector_dim)
  for ngram in generate_ngrams(word, ngrams):
  if ngram in vocabulary:
  word_vector += get_vector(ngram)
  return word_vector
function generate_ngrams(word, ngrams):
  ngrams_list = []
  for n in ngrams:
  for i in range(len(word) - n + 1):
  ngrams_list.append(word[i:i+n])
  return ngrams_list

This study utilized a pre-trained FastText model designed for the Indonesian language with a vector dimension of 300. The main parameters set for the FastText model are outlined in Table II below:

TABLE II FASTTEXT PARAMETERS

Parameter	Score
Minn	2
Maxn	5
Dim	300
Lr	0.2

Where ;

- Minn : Minimum of n-gram length,
- Maxn : Maximum of n-gram length,
- Dim : Dimensionality word vectors,
- Lr. : Learning rate.

These parameters were selected based on empirical research to optimize the performance of word embeddings for the classification task.

E. Classification

The text classification process for Indonesian will utilize one of the RNN architectures known as Long Short-Term Memory (LSTM). LSTM networks are well-suited for this task as they are capable of learning long-term dependencies in sequential data. This ability is crucial for understanding the context in which hate speech occurs, as demonstrated in Fig. 2 below.

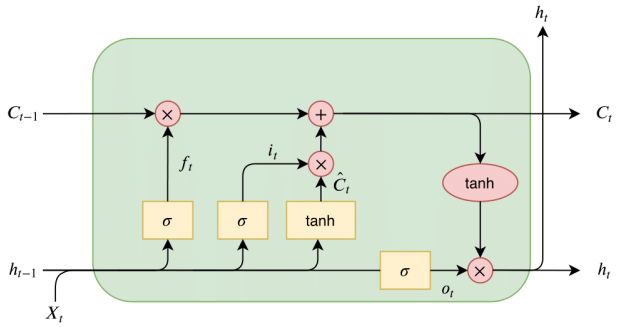


Fig 2. LSTM Architecture

At the bottom, cell gates regulate information issued to the next cell state or unit [13]. The cell state is the path at the top to send information to the next unit [14]. This mechanism enables the LSTM to determine which information to add or remove from the cell state, effectively capturing dependencies across different time steps.

F. Evaluation

The process of evaluating a system involves utilizing a confusion matrix to compute accuracy, precision, recall, and F1-score for each class. The confusion matrix serves as a tabular representation of correctly and incorrectly classified test data [15]. An illustrative example of a confusion matrix for text classification is demonstrated in Table III.

TABLE III. CONFUSION MATRIX

		Prediction Class	
		1	**0**
real class	**1**	TP	FN
	0	FP.	TN

The performance of the model is thoroughly evaluated by this confusion matrix, showcasing its precision in accurately detecting hate speech in Indonesian tweets.

IV. RESULT AND CONCLUSION

A. Result

In this study, 13,169 tweets, formatted as CSV files, were classified into two groups: HS (hate speech) and NHS (non-hate speech). The dataset consists of 5,561 tweets classified as hate speech and 7,608 tweets classified as non-hate speech. Below is an illustrative snapshot of the dataset as in Fig. 3 below:

	Tweet	HS
0	- disaat semua cowok berusaha melacak perhatia...	1
1	RT USER: USER siapa yang telat ngasih tau elu?...	0
2	41. Kadang aku berfikir, kenapa aku tetap perc...	0
3	USER USER AKU ITU AKU\n\nKU TAU MATAMU SIPIT T...	0
4	USER USER Kaum cebong kapir udah keliatan dong...	1
...	...	..
13164	USER jangan asal ngomong ndasmu. congor lu yg ...	1
13165	USER Kasur mana enak kunyuk'	0
13166	USER Hati hati bisu :(.g\n\nlagi bosan huft \...	0
13167	USER USER USER USER Bom yang real mudah terdet...	0
13168	USER Mana situ ngasih(": itu cuma foto ya kuti...	1

[13169 rows x 2 columns]

Fig. 3. Dataset Results

Fig. 3. Illustrates that the dataset contains 13,169 rows, each representing an individual tweet. Each row is composed of two primary columns: the first column contains the tweet text, which includes various expressions, phrases, and sentences as posted by users on the social media platform, and the second column contains the HS label. The HS Label is a binary classification where 1 indicates the presence of hate speech in the tweet, and 0 denotes that the tweet does not contain hate speech. This clear division enables precise categorization and facilitates practical training and evaluation of the hate speech detection model.

In order to transform raw text data into a structured format suitable for subsequent analysis, preprocessing is crucial. Cleaning, case folding, normalization, stop word removal, stemming and tokenization are the preprocessing steps undertaken in this study. Table IV provides an example of these preprocessing steps and their outputs.

The preprocessed dataset was then stored in a file named 'tweet_clean.' This file contains the cleaned, normalized, and tokenized tweet data, ready for further analysis. The Adaptive Synthetic (ADASYN) technique was used to address the issue of class imbalance, where the number of non-hate speech tweets significantly outnumbered the number of hate speech tweets.

TABLE IV. RESULTS OF PREPROCESSING DATA

Preprocessing	Input	Output
Cleaning	RT @USER @USER Udah cape gw ngasih tau elu? Kok telat bet sih??	RT USER USER Udah cape gw ngasih tau elu Kok telat bet sih
Case Folding	RT USER USER Udah cape gw ngasih tau elu Kok telat bet sih	rt user user udah cape gw ngasih tau elu kok telat bet sih
Normalisasi	rt user user udah cape gw ngasih tau elu kok telat bet sih	sudah lelah saya mamberi tau kamu kenapa telat sekali
Stopword Removal	sudah lelah memberi tau kamu kenapa telat sekali	sudah lelah memberi tau kamu kenapa telat sekali
Stemming	sudah lelah memberi tau kamu kenapa telat sekali	sudah lelah tau kamu kenapa telat kali
Tokenizing	Lelah siapa beri tau kamu kenapa telat kali	'lelah' 'beri' 'tau' 'kamu' 'telat' ''kali'

ADASYN generates synthetic data points for the minority class (hate speech) by considering the data distribution and creating new, plausible instances that help balance the dataset. This method ensures that the classification model does not become biased towards the majority class, thus promoting a more accurate and reliable detection of hate speech. Through the use of ADASYN, we achieve a balanced distribution of classes, which is crucial for the training of an effective and unbiased machine learning model.

FastText training is used to represent a word as a vector. The FastText model will be utilized as a classification model matrix. The initial step involves creating a FastText model using the dataset that has undergone preprocessing. The fastText model used is an Indonesian pre-train model with a dimension length of 300 dimensions. The fast text model does not need to be re-weighted each time the model is run, as it is saved as a file to be used repeatedly. As shown in Fig. 4 below, the model is saved in file.bin format and then converted to vector:

```
Load word embedding....
Ditemukan 3641 vector kata

Mempersiapkan embedding matrix
total kata terkonversi 3638 kata (9501 hilang)
[[ 0.          0.          0.          ...  0.          0.
   0.        ]
 [-0.26444837  0.07722441  0.10014734 ...  0.11681181 -0.13247743
  -0.08901156]
 [-0.01957436  0.17514256  0.04589798 ...  0.10342456  0.00246672
  -0.11665002]
 ...
 [ 0.          0.          0.          ...  0.          0.
   0.        ]
 [ 0.          0.          0.          ...  0.          0.
   0.        ]
 [ 0.          0.          0.          ...  0.          0.
   0.        ]]
```

Fig 4. FastText Matrix Results

After FastText training, the LSTM modeling development is carried out with the architecture, as shown in Fig 5.

```
Model: "model_LSTM"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 30, 300)           3942300
_________________________________________________________________
lstm (LSTM)                  (None, 128)               219648
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 4,162,077
Trainable params: 4,162,077
Non-trainable params: 0
_________________________________________________________________
```

Fig 5. LSTM Modeling

The LSTM architecture utilizes a sequential model to stack LSTM layers, with each LSTM layer comprising 128 cells. The optimization function employed is "adam," and the loss function is binary_crossentropy. The output assesses the accuracy of the Hate Speech classification process. The LSTM architecture undergoes training for 30 epochs, necessitating 30 complete iterations of processing the training data to achieve optimal accuracy. The batch size is set at 250, and this phase involves dataset processing using the LSTM algorithm for classification.

In this research, a direct comparison was made between two feature extraction methods, namely FastText and TF-IDF. The aim was to provide a comprehensive evaluation of the performance of FastText when it is integrated with LSTM algorithms. In the study, a split validation technique was rigorously applied to effectively split the data set into training and test data during the testing phase, with 80% being allocated to training and 20% to testing during the classification phase. After carefully implementing the LSTM algorithm, a thorough evaluation process was conducted to validate and assess the performance of this study's hate speech classification model.

The evaluation results include a comparison between the FastText-LSTM model and the TF-IDF-LSTM model. This comparison details the differences in accuracy, precision, recall, and F1 score to identify the more effective approach for hate speech detection. The outputs of the models are presented in Fig. 6 below:

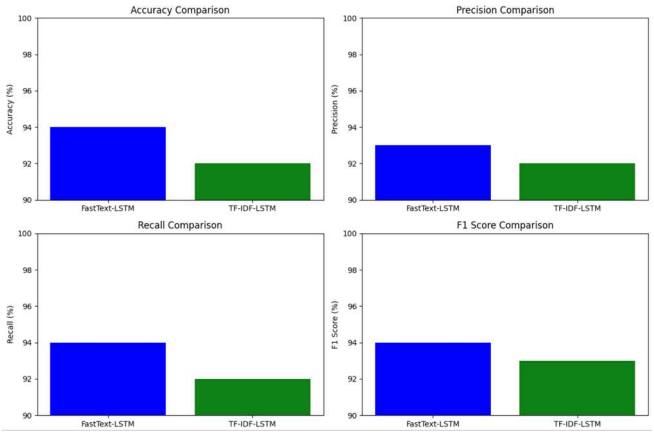
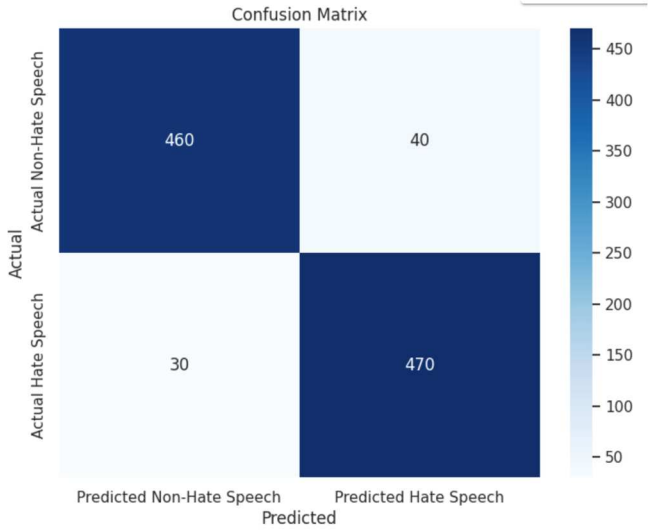


Fig 6. Comparative Performance of FastText-LSTM and TF-IDF-LSTM Models

The evaluation of the model's performance is based on accuracy, recall, and F1 score, as demonstrated in Fig. 7:



	precision	recall	f1-score	support
Non-Hate Speech	0.94	0.92	0.93	500
Hate Speech	0.92	0.94	0.93	500
accuracy			0.93	1000
macro avg	0.93	0.93	0.93	1000
weighted avg	0.93	0.93	0.93	1000

Fig 7. Confusion Matrix and LSTM Model Performance Report using FastText Model

After conducting comprehensive research, it can be concluded that the combination of FastText and Long Short-Term Memory (LSTM) networks, as implemented in this study, showed exceptional performance in detecting hate speech. The model achieved 94% accuracy, 92% precision, 94% recall, and a 93% F1 score.

B. Conclusion

The extensive research conducted in this study has shown that the combination of FastText and Long Short-Term Memory (LSTM) networks is highly effective in detecting hate speech. Several key factors contributed to this success. The use of effective pre-processing techniques, such as text normalisation, stopword removal and stemming, was crucial in preparing the raw text data for analysis. These techniques significantly improved the model's ability to accurately classify hate speech by ensuring that the data was clean and consistent. In addition, the model's performance, particularly for the morphologically rich Indonesian language, was significantly improved by the use of FastText embeddings, which can handle out-of-vocabulary words and capture subword information. The robustness of LSTM networks in maintaining long-term dependencies in sequential data was essential for understanding the context in which hate speech occurs, making LSTM networks particularly effective in this classification task.

This study highlights the importance of using sophisticated machine learning techniques to improve hate speech detection systems. The results show that combining FastText embeddings with LSTM networks improves classification accuracy and provides a robust framework for detecting hate speech in different contexts, especially for Indonesian language content. In future research, the integration of other advanced techniques and models could be further explored to improve hate speech detection systems. In addition, applying this methodology to different languages and larger datasets could validate and extend the findings of this study.

REFERENCES

[1] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, "Hate speech detection in the Indonesian language: A dataset and preliminary study," in *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Bali: IEEE, Oct. 2017, pp. 233–238. doi: 10.1109/ICACSIS.2017.8355039.

[2] F. E. Ayo, O. Folorunso, F. T. Ibharalu, and I. A. Osinuga, "Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions," *Computer Science Review*, vol. 38, p. 100311, Nov. 2020, doi: 10.1016/j.cosrev.2020.100311.

[3] B. Evkoski, N. Ljubešić, A. Pelicon, I. Mozetič, and P. Kralj Novak, "Evolution of topics and hate speech in retweet network communities," *Appl Netw Sci*, vol. 6, no. 1, Art. no. 1, Dec. 2021, doi: 10.1007/s41109-021-00439-7.

[4] S. Alashri, S. S. Kandala, V. Bajaj, R. Ravi, K. L. Smith, and K. C. Desouza, "An analysis of sentiments on facebook during the 2016 U.S. presidential election," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Aug. 2016, pp. 795–802. doi: 10.1109/ASONAM.2016.7752329.

[5] E.-J. Lee, H.-Y. Lee, and S. Choi, "Is the message the medium? How politicians' Twitter blunders affect perceived authenticity of Twitter communication," *Computers in Human Behavior*, vol. 104, p. 106188, Mar. 2020, doi: 10.1016/j.chb.2019.106188.

[6] F. A. Wenando and E. Fuad, "Detection of Hate Speech in Indonesian Language on Twitter Using Machine Learning Algorithm," *Prosiding CELSciTech*, vol. 4, pp. 6–8, 2019.

[7] Z. Mansur, N. Omar, and S. Tiun, "Twitter Hate Speech Detection: A Systematic Review of Methods, Taxonomy Analysis, Challenges, and Opportunities," *IEEE Access*, vol. 11, pp. 16226–16249, 2023, doi: 10.1109/ACCESS.2023.3239375.

[8] J. S. Malik, G. Pang, and A. van den Hengel, "Deep Learning for Hate Speech Detection: A Comparative Study," *arXiv:2202.09517 [cs]*, Feb. 2022, Accessed: Mar. 03, 2022. [Online]. Available: http://arxiv.org/abs/2202.09517

[9] S. Modha, P. Majumder, T. Mandl, and C. Mandalia, "Detecting and visualizing hate speech in social media: A cyber Watchdog for surveillance," *Expert Systems with Applications*, vol. 161, p. 113725, Dec. 2020, doi: 10.1016/j.eswa.2020.113725.

[10] S. S. Syam, B. Irawan, and C. Setianingsih, "Hate Speech Detection on Twitter Using Long Short-Term Memory (LSTM) Method," in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Nov. 2019, pp. 305–310. doi: 10.1109/ICITISEE48480.2019.9003992.

[11] M.-Y. Cheng, D. Kusoemo, and R. A. Gosno, "Text mining-based construction site accident classification using hybrid supervised machine learning," *Automation in Construction*, vol. 118, p. 103265, Oct. 2020, doi: 10.1016/j.autcon.2020.103265.

[12] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," *arXiv:1703.04009 [cs]*, Mar. 2017, Accessed: Mar. 03, 2022. [Online]. Available: http://arxiv.org/abs/1703.04009

[13] F. A. Wenando, R. Hayami, Bakaruddin, and A. Y. Novermahakim, "Tweet Sentiment Analysis for 2019 Indonesia Presidential Election Results using Various Classification Algorithms," in *2020 1st International Conference on Information Technology, Advanced Mechanical and Electrical Engineering (ICITAMEE)*, Oct. 2020, pp. 279–282. doi: 10.1109/ICITAMEE50454.2020.9398513.

[14] M. Hitesh, V. Vaibhav, Y. J. A. Kalki, S. H. Kamtam, and S. Kumari, "Real-Time Sentiment Analysis of 2019 Election Tweets using Word2vec and Random Forest Model," in *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, Sep. 2019, pp. 146–151. doi: 10.1109/ICCT46177.2019.8969049.

[15] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference*

on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[16] A. G. D'Sa, I. Illina, and D. Fohr, "BERT and fastText Embeddings for Automatic Detection of Toxic Speech," in *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, Feb. 2020, pp. 1–5. doi: 10.1109/OCTA49274.2020.9151853.

[17] A. Amalia, O. S. Sitompul, E. B. Nababan, and T. Mantoro, "An Efficient Text Classification Using fastText for Bahasa Indonesia Documents Classification," in *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, Medan, Indonesia: IEEE, Jul. 2020, pp. 69–75. doi: 10.1109/DATABIA50434.2020.9190447.

[18] G. B. Herwanto, A. Maulida Ningtyas, K. E. Nugraha, and I. Nyoman Prayana Trisna, "Hate Speech and Abusive Language Classification using fastText," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, Indonesia: IEEE, Dec. 2019, pp. 69–72. doi: 10.1109/ISRITI48646.2019.9034560.

[19] J. Kong *et al.*, "Fast Extraction of Word Embedding from Q-contexts," *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 873–882, Oct. 2021, doi: 10.1145/3459637.3482343.

[20] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[21] T. Yao, Z. Zhai, and B. Gao, "Text Classification Model Based on fastText," in *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)*, Mar. 2020, pp. 154–157. doi: 10.1109/ICAIIS49377.2020.9194939.

[22] A. G. D'Sa, I. Illina, and D. Fohr, "BERT and fastText Embeddings for Automatic Detection of Toxic Speech," in *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, Feb. 2020, pp. 1–5. doi: 10.1109/OCTA49274.2020.9151853.

[23] H. T.-T. Do, H. D. Huynh, K. Van Nguyen, N. L.-T. Nguyen, and A. G.-T. Nguyen, "Hate Speech Detection on Vietnamese Social Media Text using the Bidirectional-LSTM Model," *arXiv:1911.03648 [cs]*, Nov. 2019, Accessed: Mar. 20, 2022. [Online]. Available: http://arxiv.org/abs/1911.03648

[24] N. I. Pratiwi, I. Budi, and I. Alfina, "Hate Speech Detection on Indonesian Instagram Comments using FastText Approach," in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Yogyakarta: IEEE, Oct. 2018, pp. 447–450. doi: 10.1109/ICACSIS.2018.8618182.

[25] J. Zhang, Y. Li, J. Tian, and T. Li, "LSTM-CNN Hybrid Model for Text Classification," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Oct. 2018, pp. 1675–1680. doi: 10.1109/IAEAC.2018.8577620.

[26] T. Mandl, S. Modha, A. Kumar M, and B. R. Chakravarthi, "Overview of the HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German," in *Forum for Information Retrieval Evaluation*, in FIRE 2020. New York, NY, USA: Association for Computing Machinery, Dec. 2020, pp. 29–32. doi: 10.1145/3441501.3441517.

[27] M. Sun, I. A. Hameed, and H. Wang, "A Novel Ensemble Representation Framework for Sentiment Classification," in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, United Kingdom: IEEE, Jul. 2020, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9207194.

[28] M. O. Ibrohim and I. Budi, "Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter," in *Proceedings of the Third Workshop on Abusive Language Online*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 46–57. doi: 10.18653/v1/W19-3506.