Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 2 (2023) 302 - 309 ISSN Media Elektronik: 2580-0760

Indonesian Hate Speech Detection Using Bidirectional Long Short-Term Memory (Bi-LSTM)

Aditya Perwira Joan Dwitama[1], Dhomas Hatta Fudholi[2], Syarif Hidayat[3]
[1,2,3]Department of Informatics, Universitas Islam Indonesia
[1]aditya.dwitama@students.uii.ac.id, [2]hatta.fudholi@uii.ac.id, [3]syarif@uii.ac.id

Abstract

Social media is a platform that allows users to express themselves freely including spreading hate speech content. The government has issued the regulation in the UU ITE to handle and prevent hate speech on social media. The research was also conducted using the Bi-LSTM to classify the text into hate speech or not. Another research was purposed to detect hate speech and its categories using Bi-GRU. However, the performance of the model Bi-GRU is still lower than Bi-LSTM with an accuracy of 86.44% and 96.44%. Therefore, this study aims to build a model that can detect hate speech and its categories. The research offers Bi-LSTM as a classification model and IndoBERT as a tokenization model. The dataset used is a public dataset containing 13 thousand tweets. As a result, the best model obtained is using 20 epochs, 192 batch sizes, 1 layer Bi-LSTM with 40 nodes, and applying class weighing in the optimization process. The pre-train model from IndoBERT that is used to support the performance of the model in classifying is "indobenchmark/indobert-large-p2". The performance given by the purposed model is very good with an average accuracy, precision, and recall of 97.66%, 96.50%, and 85.25%.

Keywords: hate speech; Bi-LSTM; indoBERT; text; twitter

1. Introduction

According to Hootsuite's survey, the number of internet users in Indonesia in 2021 is 202.6 million users. As many as 83.9 percent of total internet users use the internet to access social media [1]. Social media gives users the freedom to express themselves in various ways. But not infrequently this freedom is misused to spread hate speech content [2].

Hate speech is an expression intended to demean and even discriminate against individuals and/or groups based on race, ethnicity, gender, religion, and other aspects [3]. Hate speech is dangerous for society because it can cause social conflict, discrimination, murder, and even imprisonment for perpetrators by applicable law [4]. For example, there is a text that says, "Kill them!". The text is rather a serious threat to community groups. If the group targeted by the text doesn't accept it, then social conflict will break out between the two sides.

The Indonesian government has made efforts to prevent and handle hate speech. Regulations regarding hate speech have been issued in Indonesia Information and Electronic Transactions (UU ITE) Law, chapter 28, number 2. Every citizen is prohibited from publishing anything that can incite hatred or enmity toward certain individuals or community groups based on ethnicity, religion, or race, according to its regulations [5]. However, this regulation is not enough to prevent the emergence of hate speech cases in Indonesia. During February and March 2021, the National Police's Cyber Crime Directorate issued a warning to 125 social media accounts through the virtual police because the account indicated spreading hate speech content [6].

In terms of technology, cases of hate speech have been examined to automate the identification of hate speech. Research to create a detection model utilizing the Machine Learning and Deep Learning algorithm [7], [8], [9], [10]. The studies [7], [8] discuss hate speech in Indonesian texts using LSTM. The difference between the two studies lies in the word embedding method. The study [7] uses word2vec and obtains an accuracy of 91%. Meanwhile, [8] uses FastText and gets an accuracy of 95.93%. Another study was conducted by [10] using Bi-LSTM and word2vec as word embedding methods. The study used datasets from [11] (the same as [8]) and obtained an accuracy of 94.66%. Besides being able to provide excellent performance in classifying hate speech in Indonesian text, LSTM or Bi-LSTM is also capable of classifying offensive texts in Arabic texts. The study [9] was able to provide an

Accepted: 03-11-2022 | Received in revised: 28-02-2023 | Published: 26-03-2023

accuracy of 86.4% using Bi-LSTM. However, the success of these studies is still used to classify texts into hate speech without detecting the category of hate speech.

Research on hate speech growth to be able to detect hate speech and its category [2], [3], [12], [13]. The research was conducted using Bidirectional Gated Recurrent Unit (Bi-GRU) [2]. The study use an embeddings technique consisting of word2vec, FastText, and IndoBERT. The result of the study's accuracy value while applying the IndoBERT pre-trained embedding was the best, with an accuracy of 84.77 percent [2]. However, this result is still below the accuracy of [10] which only detects text as hate speech or not.

IndoBERT is a pre-train model based on Bidirectional Encoder Representations from Transformers (BERT). It was developed specifically to handle the Indonesian language and can tokenize text at the contextual level [14]. BERT has several special tokens that can be generated in the tokenization process, namely [CLS], [SEP], and [PAD]. The [CLS] token will be added at the beginning of the text to mark the starting point of a text. The [SEP] token appears when there are two sentences in the text. The token is used as a separator between the two. The [PAD] token appears if the number of tokens generated from the tokenization process is less than the length of the token that has been defined in the BERT model. If the token length is defined as 10 and the token length is only 6, then the remaining tokens will be filled by [PAD] [15].

Various pre-train models have been developed for IndoBERT including IndoNLU [16], IndoLEM [17], and IndoBertTweet [18]. IndoNLU was constructed utilizing the Indo4B dataset, which contains approximately 250 million sentences with a total word count of 4 billion words. The IndoLEM model is constructed using a corpus dataset compiled from a variety of sources, including the Indonesian Wikipedia, news items from Kompas (https://kompas.com), Tempo (https://majalah.tempo.co), and Liputan 6 (https://liputan6.com), and the Indonesian language web corpus. Finally, there is IndoBertTweet, which uses a corpus derived directly from Twitter. IndoBertTweet uses a dataset of 409 million words with four key topics: the economy, health, education, and governance.

This study will conduct modeling to detect hate speech and its categories because there is a gap in the model's performance between hate speech only and hate speech and its categories. The model offered in this study is to combine Bi-LSTM as a classification model and IndoBERT as a tokenization model. Bi-LSTM was chosen because the development of this algorithm aims to be able to handle problems related to text analytics. Meanwhile, IndoBERT was chosen because of its ability to perform contextual embedding in text. IndoBert also got the best performance as a tokenization model in an experiment to detect hate speech [2]. Thus, this study is expected to produce a model that can detect hate speech and its categories with better performance than previous studies.

2. Research Methods

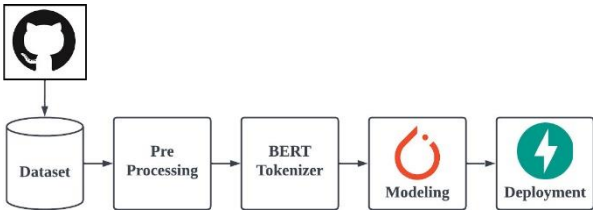

Figure 1. Research flow.

This research was conducted using Bi-LSTM as a classification model and combined with a pre-train model from IndoBERT as a tokenization model in the text. First, data exploration is carried out on the dataset that has been obtained from GitHub (https://github.com/okkyibrohim/id-multi-label-hate-speech-and-abusive-language-detection). Then, the process continues to the preprocessing stage to make the text in the dataset more standard. The standardized texts are then tokenized. The output of the tokenization produces an input vector for the Bi-LSTM model. The modeling process is the last stage in this research, followed by an evaluation of the modeling results. The flow of the research stages can be seen in Figure 1.

2.1. Dataset

Dataset used in this research is a public dataset that can be found on GitHub. The dataset is obtained from previous research [19]. The dataset contains text from Twitter and has been annotated by 30 annotators. The total data in the dataset is 13.169 tweets and assigned to 12 labels namely *hate speech, abusive, individual, group, race, religion, physical, gender, others, strong, moderate,* and *weak*. Each label is defined as a binary value with the number 1 or 0. 1 indicates that the tweet is positive to the label, whereas 0 states a negative one.

The focus of the research is on how to detect hate speech and its categories. Of the total 12 labels available, only 9 labels will be used in the research that are *hate speech, abusive, individual, group, race, religion, physical, gender,* and *others*. The selection of these labels refers to the definition of hate speech in UU ITE. The use of 9 labels is also referring to previous research [12]. The research is focused to get the best combination of labels for the hate speech detection model. The best performance is obtained when the hate speech levels (*weak, moderate,* and *strong*) are combined into one class. The model managed to get an accuracy of 68.43 percent while still describing the label under the hate speech category. Details of the hierarchy of labels used are presented in Figure 2. The

DOI: https://doi.org/10.29207/resti.v7i2.4642

labels are represented in the figure in the form of a rounded corner rectangle.

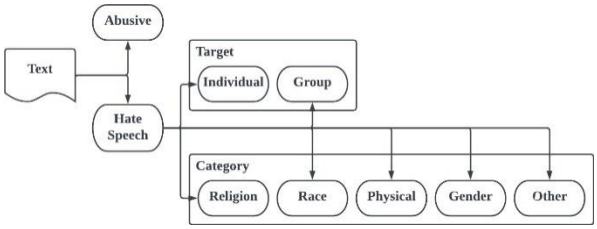

Figure 2. Label hierarchy.

2.2. Pre-processing

The text on Twitter is diverse. Even though the words they use might be the same, everyone has a different writing style. For example, some people may write a word as "tunggu" (wait). But the other one may write the word with its informal form "tgu" (w8) instead. This situation requires preprocessing to standardized text for the modeling stage.

Preprocessing is done by cleaning and normalizing the text. The cleaning process is carried out by removing unnecessary words and characters, such as numbers, punctuation, emoticons, usernames, hyperlinks, and stopwords.

The texts in the dataset are cleaned and normalized during the preprocessing stage. Unnecessary words and characters, such as digits, punctuation, emoticons, usernames, hyperlinks, and stopwords, are removed. Then normalization is done by converting the slang words into standard words. The normalization process utilizes the dictionary from the same repository as the dataset [19]. The output of this process is a collection of standard texts that are ready to be input modeling process.

2.3. Tokenization

This research will test several pre-train models from IndoBERT as a model to tokenize text. These models include *cahya/bert-base-indonesian-522M*; *indobenchmark/indobert-base-p1*; *indobenchmark/indobert-large-p2*; *indolem/indobertweet-base-uncased*; *ayameRushia/indobert-base-uncased-finetuned-indonlu-smsa*; *afbudiman/indobert-classification*

All the pre-train can be found at https://huggingface.co. As an initial setup, the research will start by using a pre-train model from *cahya/bert-base-indonesian-522M*. The model is built with a dataset sourced from the Indonesian Wikipedia with a corpus of 32,000 words.

2.3. Modeling

In this stage, the classification model will be built using Bi-LSTM architecture. The model architecture has several layers such as an embedding layer, Bi-LSTM, and a fully connected neural network. The embedding layer will act as a gateway that will receive the vectors tokenized by IndoBERT. At the end of the architecture, there is a fully connected layer neural network that acts as the output of the model. The layer consists of 9 nodes according to the number of labels used in the research. So, the model will provide output in the form of a vector with 9 elements. Each element represents the value of each label. Thus, the model architecture used can be used to categorize data multilabel [20].

The modeling process is carried out by performing hyperparameter tuning. Some of the parameters involved in this process are epoch, number of LSTM nodes, learning rate, LSTM layer, and batch size.

The modeling process also applies class weighting techniques. The application of class weighting makes the process for updating the weight will be adjusted to the conditions in each label. The condition referred to the proportion between the number of positive and negative data on each label. The equation for determining the class weight is defined in formula 1.

$$CW_i = \frac{1}{\sqrt{N_i}} \quad (1)$$

CW_i is a representation of the class weight of the i label. N_i shows the number of hate speech texts that are on i label.

2.5. Evaluation

In this research, the modeling process will apply evaluation metrics based on accuracy, precision, and recall that is calculated using formula 2, 3, and 4.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$precision = \frac{TP}{TP+FP} \quad (3)$$

$$recall = \frac{TP}{TP+FN} \quad (4)$$

The complete text that is expected to be true as hate speech is a TP (True Positive). The text that is not hate speech but is detected to be hate speech is an FP (False Positive). The text that is accurately predicted as not being hate speech is a TN (True Negative) representation. Finally, FN (False Negative) represents the text that is predicted to be non-hate speech even though they are hate speech text.

In addition, this research is applying k-fold validation in evaluation. K-fold validation works by dividing data into several groups based on k value. In this case, the number of k used is 5. The application of 5-fold in cross validation because it considers imbalanced conditions in the dataset.

2.6. Deployment

At the end of the research, a detection service is built with a Fast Application Programming Interface (API). Service API will load the best model from modeling

DOI: https://doi.org/10.29207/resti.v7i2.4642

process. It will ensure that the best model service can be used by the end user just by calling the service.

3. Results and Discussions

This section discusses the process and the results obtained during the research. Elaboration starts with data exploration and pre-processing. Then proceed with hyperparameter tuning in the modeling process. The last is making a prototype system.

3.1. Data exploration and pre-processing

Data exploration found that 146 tweets were duplicated in the dataset. Then, those tweets were eliminated to remove bias in the dataset. After the process, the dataset consists of 13.023 tweets.

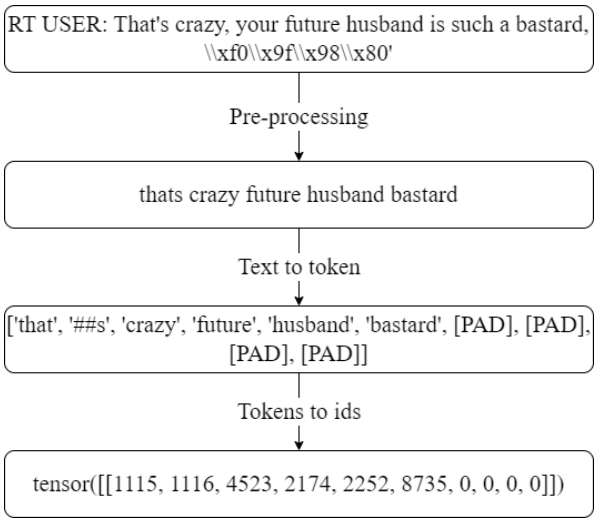

Figure 3. The illustration process of BERT tokenization.

The research continued to the pre-processing stage. The cleaning process at the pre-processing stage influences the length of the words in the tweet. Removing unnecessary characters or words reduces the length of the words in the tweet. The exploration found that the tweets in the dataset mostly have word length at number 10. The exploration result is then used as consideration in determining the maximum token in the tokenization stage. If the length of the tweet exceeds 10 then the tweet will be cut to 10 words in the tokenization process. Meanwhile, if the tweet has a word length below 10, the remaining tokens will be filled by a [PAD]. The [PAD] token on the results of tokenization by BERT has a value of 0. If the data contains many 0 values, the level of bias of the data will be higher. The condition will affect the performance of the model to classify hate speech in text. The illustration of tokenization by BERT can be seen in Figure 3.

Furthermore, to avoid large number of [PAD] token, then re-elimination is carried out on the data in the dataset. The reduction is applied to tweets with a word length equal to less than 5. After the elimination process, the dataset consists of 10.011 tweets. The distribution of positive and negative data on each label can be seen in Table 1. From the table, the dataset is imbalanced, especially on physical and gender labels. physical and gender labels have positive tweets only about 2 percent of the total tweets in the dataset.

Table 1. Distribution of positive and negative data on each label.

Label	Positive	Negative
Hate Speech	4219	5792
Abusive	3286	6725
Individual	2564	7447
Group	1655	8356
Religion	715	9296
Race	463	9548
Physical	209	9802
Other	2761	7250

3.2. Modeling and tuning

The modeling process performs hyperparameter tuning to get the best model architecture. The initial setup of the model's parameters used is epoch = 10; number of Bi-LSTM nodes = 10, learning rate = 1e-1; group size = 128; and the number of Bi-LSTM layers = 1. Class weighting is applied to the optimization process because the dataset is unbalance (Table 1). As a tokenization model, pre-train model from "cahya/bert-base-indonesian-522M" is used as the initial setup.

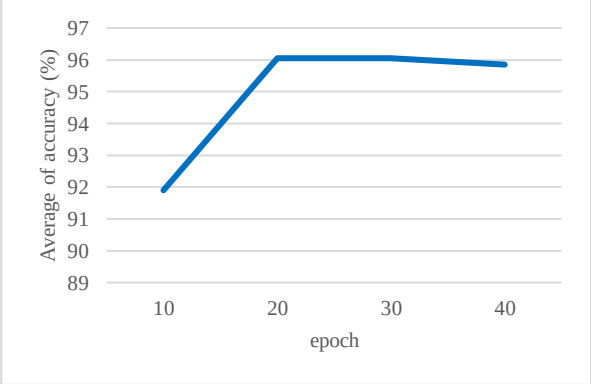

Figure 4. Comparison accuracy of the epoch experiment.

The first hyperparameter tuning that we did on the epoch. The purpose of this experiment is to see how many iterations of learning are needed by the model to get convergence. The experiment started with 10 epochs and continued with the addition of 10 epochs in the next experiment. Until the modeling process with 40 epochs, the experiment was stopped. The average accuracy of the model has increased significantly when modeling using 20 epochs. The average accuracy increased from 91.90 percent (10 epochs) to 96.05 percent (20 epochs). However, when the number of epochs was increased to 30 and 40 epochs, the average accuracy of the model did not change significantly as shown in Figure 4. Hence, we decided that the best performance of the model was obtained when modeling using 20 epochs. However, these results are still noteworthy because there are still some labels that have

DOI: https://doi.org/10.29207/resti.v7i2.4642

low recall. We present the performance of the model in Table 2.

Table 2. The model performance uses 20 epochs.

Label	Accuracy	Precision	Recall
Hate Speech	97,75%	97,39%	97,27%
Abusive	90,14%	90,49%	78,18%
Individual	93,90%	92,77%	82,61%
Group	95,24%	91,36%	78,61%
Religion	96,76%	89,49%	61,96%
Race	98,03%	93,18%	61,99%
Physical	98,87%	90,68%	**51,20%**
Other	98,96%	94,57%	**55,71%**

The second hyperparameter tuning is on the number of nodes in the Bi-LSTM layer. This experiment aims to find the optimal number of nodes to be applied to the Bi-LSTM layer. The experiment applies the number of Bi-LSTM nodes as many as 10, 20, 30, 40, 50, and 60. As a result, the two models provide the best average accuracy when using the number of nodes as much as 30 and 40. The average accuracy obtained is 96.42 percent and 96.39 percent as shown in Table 3. The margin of the average accuracy of both models is only 0.03 percent.

Table 3. Comparison accuracy of the experiment of Bi-LSTM nodes

Number of nodes	Accuracy
10	96,05%
20	96,08%
30	96,42%
40	96,39%
50	95,86%
60	94,96%

We conduct further analysis to find the best model by performance on recall as shown in Table 4. recall was chosen because the performance of the model in this matrix is still low compared to precision and accuracy (see Table 2). Based on Table 4, The model that applies the number of Bi-LSTM nodes as much as 40 provides better recall performance. *Physical* and *gender* labels managed to penetrate the recall value above 60 percent. Hence, we decide to choose the best model when applying 40 nodes in Bi-LSTM layers.

Table 4. Comparison of recall between Bi-LSTM 40 and 30 nodes.

Label	40 node	30 node
Hate Speech	97,18%	97,23%
Abusive	78,00%	81,95%
Individual	85,14%	83,50%
Group	79,34%	78,61%
Religion	71,33%	68,39%
Race	75,81%	73,87%
Physical	**68,42%**	**59,81%**
Other	**67,12%**	**59,36%**

The second hyperparameter tuning is on learning rate. In the experiment on the number of Bi-LSTM nodes, the learning rate used is 1e-1. The average accuracy obtained by the model is 96,39 percent. The experiment reduced the learning rate to a tenth of the initial value. The lower the learning rate, the lower the model's performance. The average accuracy when using a learning rate of 1e-2 dropped to 94.47 percent. The performance is getting lower when using 1e-3. The average accuracy obtained is only 82.14 percent. The experimenter then decided to increase the learning rate from the initial setup to 5e-1. The average accuracy obtained is also lower at 83.65 percent. Hence, the experiment decided that the best model was obtained when using a learning rate of 1e-1. The result of this experiment can be seen in Figure 5.

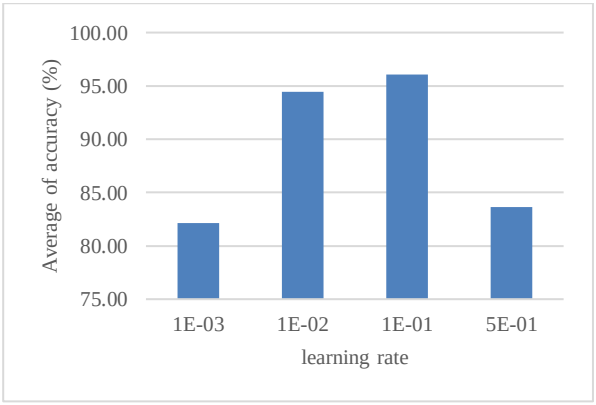

Figure 5. The experiment result of tunning on learning rate.

The fourth hyperparameter tuning is on batch size. The experiment tested 8 variations of batch size during modeling, namely 128, 32, 64, 96, 160,192, 224, and 256. The result of this experiment can be seen in Table 5. The Model performance decreased when the batch size was reduced to 96, 64, and 32. But it's different when the batch size is increased from the initial setup. The model can obtain an average accuracy of above 97 percent when using batch sizes 160, 192, 224, and 256.

Table 5. The experiment result of tunning on batch size.

Batch Size	Accuracy
32	89,15%
64	93,15%
96	95,08%
128	96,05%
160	97,27%
192	97,21%
224	97,62%
256	97,13%

The recall value of the batch size variation was also analyzed to determine the best model for this experiment. Based on Table 6, batch size 192 and 224 manage to provide the best average recall. However, batch size 192 has a better recall to correctly predict *gender* labels. the recall of the gender label in the previous experiment was below 70 percent (Table 4), in batch size 192 it was successfully increased to 76,71 percent. Hence, a batch size of 192 was decided as the best model in this experiment.

On top of tuning the above hyperparameters, we also conducted experiments by comparing different pre-trained models of IndoBert. There are 6 pre-train models of IndoBert compared in this experiment. Table 7 show the comparison of result using variation

DOI: https://doi.org/10.29207/resti.v7i2.4642

IndoBERT model as a tokenizer. The performance of each model has almost the same average accuracy. The highest performance is obtained when tokenization using the pre-train model from *indolem/indobertweet-base-uncased*. The model gets an average accuracy of 97.84 percent and is only 0.18 percent different from the *indobenchmark/indobert-large-p2* model. If we look at the recall (Table 8), the pre-train model from *indobenchmark/indobert-large-p2* shows a higher average recall of 85.25 percent. The model is also able to improve recall performance in each label. The recall value of each label managed to pass 80 percent with only *gender* at 72.15 percent. Hence, this experiment decided that the model got better performance when using a pre-train model from *indobert-large-p2*.

Table 6. Comparison of recall between variation batch size.

Labels	Batch Size			
	160	192	224	256
Hate Speech	94,29%	97,06%	99,05%	97,63%
Abusive	71,58%	81,38%	85,03%	83,38%
Individual	76,33%	88,18%	90,02%	83,39%
Group	70,82%	83,14%	86,16%	79,52%
Religion	58,14%	78,04%	79,02%	80,56%
Race	64,80%	78,62%	81,64%	84,88%
Physical	55,98%	65,17%	69,38%	68,42%
Gender	48,40%	**76,71%**	66,67%	66,67%
Other	85,44%	92,47%	94,75%	93,15%
Average	**69,53%**	**82,31%**	**83,52%**	**81,96%**

Table 7. Comparison of IndoBERT performance as a tokenizer model.

Model	Accuracy
cahya/bert-base-indonesian-522M	96,05%
indobenchmark/indobert-base-p1	97,53%
indobenchmark/indobert-large-p2	**97,66%**
indolem/indobertweet-base-uncased	**97,84%**
ayameRushia/indobert-base-uncased-finetuned-indonlu-smsa	97,46%
afbudiman/indobert-classification	96,89%

Table 8. Comparison recall between pre-train model IndoBERT.

Label	indobenchmark/indobert-large-p2	indolem/indobertweet-base-uncased
Hate Speech	98,13%	85,40%
Abusive	84,08%	90,60%
Individual	88,73%	85,56%
Group	84,83%	80,84%
Religion	81,96%	82,72%
Race	80,78%	74,64%
Physical	82,78%	73,97%
Gender	72,15%	94,17%
Other	93,77%	85,40%
Average	85,25%	85,19%

Furthermore, we conducted more experiments by increasing the number of Bi-LSTM layers, removing the class weighting, and changing the model architecture to LSTM (removing bidirectional). The results presented in Table 9 show that the additional experiment reduces the model's performance. Removing class weights in the optimization process makes label recall drop drastically from 85,25 to 52,35 percent. The same thing happens when the complexity of the model is increased by using 2 layers of Bi-LSTM. The average recall obtained is very low only 36,26 percent.

Table 9. Comparison of the best model with additional experiments.

Model	Akurasi	Presisi	Recall
Bi-LSTM (2 layers) + class weighting	91,16%	82,86%	35,26%
Bi-LSTM (1 layer)	98,06%	95,36%	80,35%
LSTM (1 layer) + class weighting	92,07%	87,81%	52,35%
Purposed Model Bi-LSTM (1 layer) + class weighting	97,66%	96,50%	85,25%

From the modeling stage and the hyperparameter tuning process, the best performance for detecting hate speech and its categories was obtained when using the following hyperparameters: epochs = 20, batch size = 192, learning rate = 1e-1, Bi-LSTM layer = 1, number of nodes = 40, and class weighting. The best pre-trained model for tokenization is from "indobenchmark/indobert-large-p2". The model managed to provide an accuracy of 97,66 percent. The model's performance is also supported by an excellent AUC score. In Table 10, each label gets an AUC score above 99% with an average of 99.80%.

Table 10. AUC Score of the purposed model.

Label	AUC Score
Hate Speech	99,93%
Abusive	99,48%
Individual	99,71%
Group	99,70%
Religion	99,83%
Race	99,86%
Physical	99,98%
Other	99,96%
Average	99,78%

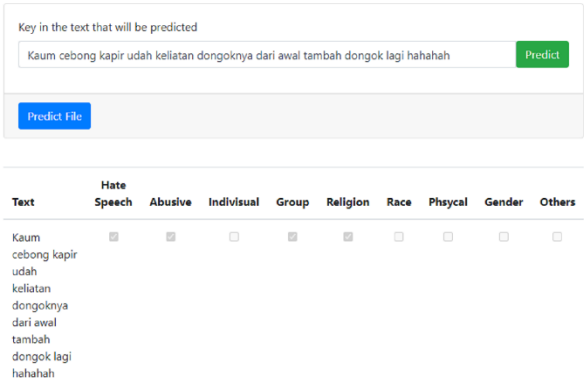

Figure 6. The user interface of the hate speech detector.

3.3. Deployment

The final stage of this research is to make a prototype of the use of the model in the form of a webpage. FastAPI is used to build services to access the model and make predictions from the client side. So, the client can directly get the prediction or detection results from a text by calling the API endpoint that has been built. It can be seen in Figure 6 that the prediction results were

DOI: https://doi.org/10.29207/resti.v7i2.4642

given to 9 labels. This is in accordance with the research concept to detect hate speech and its category from the given input.

3.4. Discussions

The imbalance dataset issue in the multi-label hate speech dataset can be overcome by using class weighting. The model manages to provide excellent performance in detecting hate speech and its category in Indonesian tweets with an accuracy of 97.66 percent (Table 9). This result is better when compared to the model that does not apply class weighting. A significant margin can be seen in the recall value with the difference of 5.15 percent.

The experiment tried to carry out the bidirectional from the model. Surprisingly, when bidirectional architecture is carried out, the model performance is degrading. The Indonesian language can be said to be a language that has simple characteristics. It does not have many tenses as English. Indonesians have the same structure regardless of the adverb. The model without the bidirectional concept has a performance accuracy of 98,06 percent but is worst in recall at only 52,35 percent (Table 9). This means that the model has a very poor performance in recognizing texts that are positive hate speech.

The results of this study can be aligned with previous studies. When compared, the average accuracy obtained can still compete with the accuracy of previous studies. The details of the alignment can be seen in Table 11.

Table 11. Research performance on hate speech detection models in text.

Reference	Dataset	Is Multilabel	Model	Acc (%)
[7]	Indonesian	No	LSTM	91,00
[10]	Indonesian	No	Bi-LSTM	94,66
[8]	Indonesian	No	LSTM	97,39
[2]	Indonesian	Yes	Bi-GRU	84,70
[9]	Arabic	No	CNN	87,40
[14]	English	No	CNN	82,00
Purposed model	**Indonesian**	**Yes**	**Bi-LSTM**	**97,66**

However, the purposed model is struggling to improve the recall on some labels. We can refer to Table 8 where the recall of *gender* is 71,15 percent while the other labels had recall values above 80 percent. This is probably because *gender* label only has about 2 percent of positive data in the dataset. Some examples of tweets that fail to be detected as hate speech *gender* can be seen in Table 12. Tweets 1 and 3 contain the words “banci” (sissy) that offend gender. The word “banci” in tweets that are positive for *gender* appears around 70 times. That’s made it become the most often word that appears in positive hate speech *gender*. However, the model failed to detect the tweets as hate speech *gender*.

The misclassifying may be due to the tokenization process. In the modeling process, the model is designed to accept input with a vector length of 10. Tweets 1 and 3 have a word length of less than 10 after preprocessing. This causes the number of tokens in the text to be less than 10. Tokens that are still empty are finally filled by tokens [PAD] from BERT so that the tokens become full. This condition makes the model wrong in predicting the output of the model because in the learning process there may be a lot of negative data that also contain [PAD] token.

Table 12. Misclassifying of hate speech *gender*.

Text	Result
*USER Tangkap aja itu jendral banci kayak *** yg bodoh* (Catch the sissy general like the stupid ***)	Expected: *Hate speech, Abusive, Individual, Gender* Actual: *Non-Hate Speech*
USER Bu guru enakan jadi jablay atau guru esde sih.\nKayaknya menikmati jadi pecun ini guru (Is it better for you to be a rude or an elementary school teacher. I think this teacher enjoys being a loser)	Excpected: *Hate speech, Abusive, Individual, Gender* Actual: *Non-Hate Speech*
Jadi cowo itu harus Gantle kalo ga Gantle itu namanya BANCI (To be a boy, you must be gentle if you're not gentle, it's called a sissy)	Expected: *Hate speech, abusive, Gender* Actual: *Non-Hate Speech*

Meanwhile, for tweet number 2, the annotator might classify the tweet as hate speech “gender” because there are the words "jablay" (girlish). However, the word only appears about 8 times in positive data in *gender*. This possibility makes the model fail to detect gender-labeled hate speech in tweets. This condition may make the model fail to detect hate speech *gender* in the tweet.

4. Conclusion

The best model for detecting hate speech and its category was successfully obtained 20 epochs, 192 batch sizes, learning rate 1e-1, 40 nodes Bi-LSTM layer, and applying class weighting. The architecture of the model utilizes the pre-train model of “indobenchmark/indobert-large-p2” as a model tokenize. The performance of the purposed model is excellent with an average accuracy of 97.66 percent.

The challenge of this research is the difficulty of increasing the performance of recall, especially on the *gender*. The author suggests doing further research on how to improve the performance of the model so that the recall value generated can be equivalent to the other 2 classification matrices’ accuracy, and precision.

Acknowledgment

We would like to say thank you to Okky Ibrahim who already permitted us to use his dataset as the basis for our proposed model in this research.

References

[1] “Digital 2022: Indonesia — DataReportal – Global Digital Insights.” https://datareportal.com/reports/digital-2022-indonesia (accessed Oct. 02, 2022).

DOI: https://doi.org/10.29207/resti.v7i2.4642

[2] A. Marpaung, R. Rismala, and H. Nurrahmi, "Hate Speech Detection in Indonesian Twitter Texts using Bidirectional Gated Recurrent Unit," in *KST 2021 - 2021 13th International Conference Knowledge and Smart Technology*, Jan. 2021, pp. 186–190. doi: 10.1109/KST51265.2021.9415760.

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

[4] C. Febriyani, "The Danger of Hate Speech in Cyberspace is Regulated as a Crime in UU ITE (Bahaya Ujaran Kebencian di Dunia Maya Diatur Sebagai Tindak Pidana di UU ITE)," 2021. https://www.industry.co.id/read/93219/bahaya-ujaran-kebencian-di-dunia-maya-diatur-sebagai-tindak-pidana-di-uu-ite (accessed Feb. 24, 2022).

[5] D. Putri, "Should all hate speech be punished? Notes for revision of UU ITE (Apakah semua ujaran kebencian perlu dipidana? Catatan untuk revisi UU ITE)," 2021. https://theconversation.com/apakah-semua-ujaran-kebencian-perlu-dipidana-catatan-untuk-revisi-uu-ite-156132 (accessed Feb. 24, 2022).

[6] A. P. J. Dwitama, "Hate Speech Detection on Indonesian Twitter using Machine Learning: Review Literature (Deteksi Ujaran Kebencian Pada Twitter Bahasa Indonesia Menggunakan Machine Learning: Reviu Literatur)," *Jurnal SNATi*, vol. 1, pp. 31–39, 2021.

[7] A. S. Saksesi, M. Nasrun, and C. Setianingsih, "Analysis Text of Hate Speech Detection Using Recurrent Neural Network," in *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2018, pp. 242–248. doi: 10.1109/ICCEREC.2018.8712104.

[8] E. Sazany and I. Budi, "Deep Learning-Based Implementation of Hate Speech Identification on Texts in Indonesian: Preliminary Study," in *2018 International Conference on Applied Information Technology and Innovation (ICAITI)*, Sep. 2018, pp. 114–117. doi: 10.1109/ICAITI.2018.8686725.

[9] H. Mohaouchane, A. Mourhir, and N. S. Nikolov, "Detecting Offensive Language on Arabic Social Media Using Deep Learning," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Oct. 2019, pp. 466–471. doi: 10.1109/SNAMS.2019.8931839.

[10] A. R. Isnain, A. Sihabuddin, and Y. Suyanto, "Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 2, p. 169, Apr. 2020, doi: 10.22146/ijccs.51743.

[11] 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 2017*, May 2018, vol. 2018-January, pp. 233–237. doi: 10.1109/ICACSIS.2017.8355039.

[12] F. A. Prabowo, M. O. Ibrohim, I. Budi, and Institute of Electrical and Electronics Engineers, "Hierarchical Multi-label Classification to Identify Hate Speech and Abusive Language on Indonesian Twitter," in *2019 6th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, 2019. doi: 10.1109/ICITACEE.2019.8904425.

[13] M. O. Ibrohim, M. A. Setiadi, and I. Budi, "Identification of hate speech and abusive language on Indonesian twitter using theword2vec, part of speech and emoji features," in *Advanced Information Science and System*, Nov. 2019. doi: 10.1145/3373477.3373495.

[14] P. Malik, A. Aggrawal, and D. K. Vishwakarma, "Toxic Speech Detection using Traditional Machine Learning Models and BERT and fastText Embedding with Deep Neural Networks," in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, Apr. 2021, pp. 1254–1259. doi: 10.1109/ICCMC51019.2021.9418395.

[15] S. Agarwal and C. R. Chowdary, "Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19," *Expert Syst Appl*, vol. 185, Dec. 2021, doi: 10.1016/j.eswa.2021.115632.

[16] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, Sep. 2020, pp. 843–857. Accessed: May 23, 2022. [Online]. Available: https://aclanthology.org/2020.aacl-main.85

[17] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," in *Proceedings of the 28th International Conference on Computational Linguistics*, Nov. 2020, pp. 757–770. doi: 10.18653/v1/2020.coling-main.66.

[18] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Sep. 2021, pp. 10660–10668. doi: 10.18653/v1/2021.emnlp-main.833.

[19] 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*, 2019, pp. 46–57. [Online]. Available: https://www.komnasham.go.id/index.php/

[20] F. Herrera, F. Charte, A. J. Rivera, and M. J. del Jesus, "Multilabel Classification," in *Multilabel Classification*, Springer International Publishing, 2016, pp. 17–31. doi: 10.1007/978-3-319-41111-8_2.

DOI: https://doi.org/10.29207/resti.v7i2.4642