Syntax Literate: Jurnal Ilmiah Indonesia P–ISSN: 2541-0849
e-ISSN: 2548-1398
Vol. 7, No. 2, Februari 2022

# DETECTING HATE SPEECH IN TWITTER USING LONG SHORT-TERM MEMORY AND NAÏVE BAYES METHOD

**Firman Sriyono, Kusrini, Asro Nasiri**
AMIKOM University Yogyakarta, Indonesia
Email: firman.1292@students.amikom.ac.id, kusrini@amikom.ac.id, asro.nasiri@amikom.ac.id

## Abstract

The information technologi's development has been very sophisticated and easy, so that it becomes a lifestyle for people throughout the world without exception Indonesia which also affected by the development of this technology. One of the benefits of information technology is the emergence various kinds of social networking sites or social media such as Facebook, Twitter and Instagram. Technological developments isn't only have a positive impact, but also have a negative impact the crime of insult or hate speech. This study is aims to classify Indonesian hate speech sentences based on hate speech and neutral sentiments using the Long Short-Term Memory (LSTM) method. Research data is obtained from Indonesian-language tweets. In testing process, the LSTM method will be compared with the Naïve Bayes method.

**Keywords**: hate speech; hate speech detection; long short-term memory; abusive language; sentiment analysis; naïve bayes

Received: 2022-01-20; Accepted: 2022-02-05; Published: 2022-02-20

## Introduction

The development of information technology is very sophisticated and easy, so that it becomes a lifestyle for people around the world, without exception Indonesia is also affected by the development of this technology. The population of Indonesia which always increases every year because the birth rate continues to increase, so the use of technology is needed to support daily activities. One of the benefits of information technology is the emergence of various kinds of social networking sites or social media, users of this social networking site or social media cover various groups ranging from children, students, housewives, traders, employees and so on. Social media is widely used by the people of Indonesia and we can find it through search engines such as Google or Mozilla Firefox, but the most popular among social media users are Facebook, Twitter, BBM, WhatsApp, Instagram. Legal problems that are often faced are related to the delivery of information and communication, especially in terms of evidence and matters related to law which are implemented through electronic systems. As a result of these developments, information technology by itself has also changed the behavior of people from global human civilization. But technological developments not only have a

**How to cite:** Sriyono. F., Kusrini & Asro Nasiri (2022) Detecting Hate Speech In Twitter Using Long Short-Term Memory And Naïve Bayes Method. *Syntax Literate: Jurnal Ilmiah Indonesia, 7*(2). http://dx.doi.org/10.36418/Syntax-Literate.v7i2.6313
**E-ISSN:** 2548-1398
**Published by:** Ridwan Institute

positive impact, but also have a negative impact criminal acts of humiliation or hate speech and the dissemination of information that aims to cause hatred or hostility between certain individuals or groups based on ethnicity, religion and race (Alfina, Sigmawaty, Nurhidayati, & Hidayanto, 2017).

Hate speech is an act of communication carried out by an individual or group in the form of provocation, incitement, or insult to another individual or group in terms of various aspects such as race, color, ethnicity, gender, disability, sexual orientation, nationality, religion, and so on. In a legal sense, Hate speech is a word, behavior, writing, or performance that is prohibited because it can trigger acts of violence and prejudice either on the part of the perpetrator or the victim of the act (Waseem & Hovy, 2016).

One way to reduce hate speech on Twitter is to add a filter to the tweet section. While the easy way to stop hate speech on Twitter is to block tweets for every syllable to sentences that are considered hate speech (Zhang, Beetz, & de Vries, 2018).

The hate speech detection method proposed by (Alshalan & Al-Khalifa, 2020) uses CNN (Convolutional Neural Network), GRU (Gated Recurrent Units) and BERT (Bidirectional Encoder Representations From Transformers) methods to detect hate speech. This method has sufficient performance but the accuracy produced is felt to be lacking and also the semantic features between words are rarely considered in text classification. The LSTM method has been used by (Artur, 2021) and has good results compared to conventional methods. The LSTM method proves that it is suitable for text classification.

We takes the case of detecting hate speech because there are many people who are not responsible for using twitter as a medium to spread hate speech. Hate speech cases in Indonesia are mushrooming and rampant and disturbing the general public. This has made the world of social media polluted by the actions of these individuals.

Based on the background described, this study uses a deep learning approach to detecting hate speech on Twitter using the Long Short-Term Memory and Naïve Bayes method and also Word2vec as feature extraction.

The method for detecting hate speech was proposed by (Alshalan & Al-Khalifa, 2020) using the CNN (Convolutional Neural Network), GRU (Gated Recurrent Units) and BERT (Bidirectional Encoder Representations From Transformers) methods to detect hate speech. From the research that has been carried out, the results show that the CNN method successfully outperformed the GRU method, with an F1 score of 0.79 and AUROC 0.89. The results also showed that the BERT method failed to improve baseline results and other evaluated methods.

Another method of detecting hate speech was proposed by (Burnap & Williams, 2014) using the DNN (Deep Neural Network) and CNN (Convolutional Neural Networks) methods. From the research conducted, the results show that in general, most of the results obtained have additional low uniqueness scores. This pattern is particularly strong in the WZ and WZ, pj datasets, where most of the correct data positions added have very low uniqueness scores. Most of the data (between 50 and 60%) have u (ti) = 0, suggesting that the tweet does not have any words indicated by Hate speech.

Another method of detecting hate speech was proposed by (Mutanga, Naicker, & Olugbara, 2020) which uses the SVM (Support Vector Machine) method. The results showed that the meta classifier had a 4-gram character weight and the unigram word as the highest contributor to the overall score. 4-grams like "jew", "ape", "mud", "egro" are one of the strongest signals of hate speech. Unigram's features such as "invasion" and "violence" contribute highly to the classification of hate speech, and appear to fall under the category of hate speech. The study found that the accuracy of all display classifiers was at least 2% lower.

Another hate speech detection method proposed by (V. Pathak, M. Joshi, P. Joshi, M. Mundada, 2020) explains that the BERT (Bidirectional Encoder Representations From Transformers) Method which is included in the proposed NLP (Natural Language Processing) is compared with the XLNet, RoBERTa and LSTM methods. The study divided the datasets with a ratio of 80:20 each for training and model testing. The results showed that DistilBERT (distilbert-base-uncased) recorded an F-measure score of 75%, while LSTM with attention recorded the lowest F-measure score of 66%. Although DistilBERT has fewer layers and parameters, it excels in all other transformer algorithms explored in this study.

The hate speech method research conducted by (S. Biere, 2018) uses Natural Language Processing (NLP) and Machine Learning (ML) methods. The results of his research resulted in a model that predicts each category with an accuracy of 91% and a loss of 36%. The latter model gives an overall precision of 0.91, a gain of 0.90 and an F1 score of 0.90. The study observed that the overall model did not identify some tweets as hate speech tweets and almost 80% of the tweet data were classified as hate speech.

## Method

In this part, our discussion is how to create dataset and the methodology in conducting hate speech detection.

### A. The Dataset

The source of the dataset is obtained from Twitter and collects Indonesian-language tweets by crawling and utilizing the Rstudio application. Tweets collected are related to the presidential election in Indonesia. Dataset was carried out during the Indonesian presidential election from April 17 to August 20, 2021. This event has the potential to be a source of hate speech data because there are many pros and cons among millions of people.

**Table I**
**Twitter Data Collection**

| Tweet | Indicators |
|---|---|
| Siapapun Cawapresnya, tetap dukung Pak #Jokowi2Periode | Neutral |
| Pdkng jenderal kardus n jenderal baper kampret ngak malu bahas hutang lah sang jenderal sj byk hutang pribadi #Hatespeech | Hate speech |
| Kalau benar terbukti ttg mahar 500M ini. Pasangan ‘Duo Kardus’ ini gak layak jadi kontestan pilpres #TurunkanJokowi | Hate speech |

Some of the keywords used are #TurunkanJokowi, #Hate speech, #Jokowi2Periode etc. We managed to collect 4,500 tweets. After subtracting duplicate tweets we get 3,000 data which will be labeled. Distribution of hate speech data that has been labeled for each sentiment is shown in table I.

**Table 2**
**Sentiment Data Distribution**

| Sentiment | Data amount |
|---|---|
| Hate speech | 1920 |
| Neutral | 1080 |

## B. Detecting Hate Speech

The purpose of this study is to compare the features and methods used to determine the combination of features that have the best performance. The methods we work on consist of: 1) preprocessing; 2) feature extraction; 3) classification; and 4) comparison method results.

1. Preprocessing

We modeled the preprocessing (Burnap & Williams, 2014) method by adding minor modifications to the flow. The preprocessing steps we use are:

a) *Case Folding*

Is step to change the letters in the comments to lowercase characters.

**Table 3**
**Case Folding Stage**

| Input | Output |
|---|---|
| Siapapun Cawapresnya, tetap dukung Pak #Jokowi2Periode | siapapun cawapresnya, tetap dukung pak #jokowi2periode |

b) *Normalization Feature*

Is step to remove special characters in comments like: period (.), comma (,), question mark (?), exclamation mark (!) and etc.

**Table 4**
**Normalization Stage**

| Input | Output |
|---|---|
| siapapun cawapresnya, tetap dukung pak #jokowi2periode | siapapun cawapresnya tetap dukung pak jokowi2periode |

c) *Stop Word Removal*

Is a stopword removal process. Stopwords are words that often appear in documents but the meaning of these words is not descriptive. For example "at", "by", "on", "a", "because" and so on;

**Table 5**
**Stop Word Result**

| Input | Output |
|---|---|
| siapapun cawapresnya, tetap dukung pak #jokowi2periode | siapapun cawapresnya dukung pak jokowi2periode |

d) *Retweet Removal*

This step is to delete identical tweets that appear in the dataset;

e) *Slangwords*

Is the process of converting non-standard words into standard words. This step is carried out using the help of a slangword dictionary and also the equivalent in standard words.

**Table 6**
**Slangword Result**

| Input | Output |
|---|---|
| klw bukan presiden yg bekerja buat rakyat siapa lagi klw bukan jokowi gak bakalan ada yg mau | Jika bukan Presiden yang bekerja buat rakyat siapa lagi jika bukan jokowi tidak akan ada yang mau |

The preprocessing (Tripathy, Agrawal, & Rath, 2016) steps that we don't use are negation handling and hashtag handling.

2. Feature Extraction

Our research uses the *Bag of Word Vector* (BoWV) and *word2vec* with the aim of representing text. The features used are bigram and trigram.

**Table 7**
**Bigram Result**

| Tweet Data | Bigram |
|---|---|
| siapapun cawapresnya dukung pak jokowi2periode | siapapun cawapresnya |
| | cawapresnya dukung |
| | dukung pak |
| | pak jokowi2periode |

For bigram implemented n=2 and for trigram implemented n=3.

**Table 8**
**Trigram Result**

| Tweet Data | Trigram |
|---|---|
| siapapun cawapresnya dukung pak jokowi2periode | siapapun cawapresnya dukung |
| | cawapresnya dukung pak |
| | dukung pak jokowi2periode |

The use of the n-gram character comes from (Schmidt & Wiegand, 2019). So our research uses 2 features, bigram and trigram. Implementation of Bag of Word Vector feature in this case can be implemented as follows:

(1) siapapun cawapresnya dukung pak jokowi2periode

(2) mana lebih menjawab rasa keadilan

Based on these two sentences, a list is made as follows for each document:

“siapapun“, “cawapresnya“, “dukung“, “pak“, “jokowi2periode“.

“mana“, “lebih“, “menjawab“, “rasa“, “keadilan“.

Represents Each Bag Of Words Vector As A JSON Object, And Associates Each Variable:

BOWV1={“siapapun“:1, “cawapresnya“:1, “dukung“:1, “pak“:1, “jokowi2periode“:1};

BOWV2={“mana“:1, “lebih“:1, “menjawab“:1, “rasa“:1, “keadilan“:1};

If the two sentences are combined it will become:

(3) siapapun cawapresnya dukung pak jokowi2periode. mana lebih menjawab rasa keadilan.

The sentence representation will be:

BOWV3={“siapapun“:1, “cawapresnya“:1, “dukung“:1, “pak“:1, “jokowi2periode“:1, “mana“:1, “lebih“:1, “menjawab“:1, “rasa“:1, “keadilan“:1};

For This Case, We Can Create Two Lists To Record The Frequency Terms Of All The Different Words.

(1) [1, 1, 1, 1, 1]

(2) [1, 1, 1, 1, 1]

3. Classification

   Our research uses a supervised learning approach to detect hate speech in Indonesian. we will test the naive bayes algorithm using existing dataset. This research uses precision, recall and f-measure validation methods for all classes.

4. Comparison of method result

   In this study, we will compare the performance and results of the Long Short-Term Memory (LSTM) method with the Naive Bayes method by utilizing the *sklearn library* in python programming language to find out the best results from the classification of hate speech.

## Result and Discussions

In this point, the results of the analysis and testing experiments will be carried out.

### A. Experiment Result

Experiments were carried out to find the best model in recognizing sentiments on hate speech. Tests are carried out to find the parameters that produce the best accuracy from the LSTM architecture created. The parameters and values tested can be seen in Table IX. In addition, two word2vec architectures will be tested, CBOW (Continuous bag-of-word) architecture and the skip-gram architecture.

**Table 9**
**Tested Parameter**

| Epoch | Activation Function |
|---|---|
| 50 | Tanh |
| 75 | Sigmoid |
| 100 | Relu |

## B. LSTM Testing

### 1. Word2vec Testing

This test begins with testing word2vec. Skipgram architecture and Continuous bag-of-word (CBOW). This test uses the LSTM method with other parameters chosen at random. The best architecture of the word2vec model will be used in the next test. The results of the word2vec test are shown in Table X.

**Table 10**
**Testing Word2vec On LSTM**

| Word2vec Architecture | Time (minutes) | Accuracy (%) |
|---|---|---|
| Skipgram | 20.15 | 72.05 |
| CBOW | 22.25 | 66.55 |

In Table X explained that the *skipgram* architecture has better accuracy than the CBOW architecture. This is because the skipgram architecture can produce better word embedding so that it can increase accuracy.

### 2. Epoch Testing

The next test is the epoch by determining the number of epochs to be tested. In this study the number of epochs tested were 50, 75 and 100. In Table 6.4 it is explained that the accuracy of epoch 50 is the best accuracy with 85.31%, but when the epoch is changed to 75 the accuracy results decrease by 84.33%. While in epoch 100 also decreased with an accuracy of 83.06%. the number of epoch tests is shown in Table XI.

**Table 11**
**Epoch Test**

| Epoch | Time (minutes) | Accuracy (%) |
|---|---|---|
| 50 | 5.45 | 85.31 |
| 75 | 6.32 | 84.33 |
| 100 | 8.04 | 83.06 |

### 3. Activation Function Test

The next test is testing the activation function. In this study, the activation function that will be tested is tanh, sigmoid, and relu. The other parameters are taken from the parameter values that produce the best accuracy in the previous test epoch 50. The results of testing the activation function can be seen in Table XII.

**Table 12**
**Activation Function Test**

| Activation Function | Time (minutes) | Accuracy (%) |
|---|---|---|
| Sigmoid | 9.09 | 84.39 |
| Tanh | 11.55 | 65.65 |
| Relu | 14.25 | 73.33 |

From the tests conducted, it shows that the sigmoid activation function produces the best accuracy value of 84.39%. Compared with the tanh activation function with an accuracy of 73.33% and the relu activation function with an accuracy of 65.65%.

### 4. LSTM Test Results

After testing several predetermined parameters the word2vec architecture, the number of epochs, and the activation function, the overall test results can be seen in Table XIII.

**Table 13**
**Lstm Test Results**

| Parameter | Value | Accuracy (%) |
|---|---|---|
| Word2vec Architecture | *Skipgram* | 84.39% |
| *Epoch* | 50 | |
| Activation Function | Sigmoid | |

### 5. Naïve Bayes Experiments and Testing Results

Testing with the Naïve Bayes method uses the same data as the data used in the Long Short-Term Memory method. The data has also gone through the same preprocessing process. The results of testing the Naïve Bayes method can be seen in table XIV (Turgut, Aydin, & Sertbas, 2016).

**Table 14**
**Naïve Bayes Test Results**

| Method | Time (minutes) | Accuracy (%) |
|---|---|---|
| Naïve Bayes | 5.20 seconds | 55.18 % |

### 6. Sentiment Results Comparison

The comparison of the accuracy results from the classification test using the long short-term memory method and the Naïve Bayes method is shown in Table XV. Classification accuracy results show the method of long short-term memory produces better accuracy than Naïve Bayes methods.

**Table 15**
**Comparison Of Classification Accuracy Results**

| Method | Time (minutes) | Accuracy (%) |
|---|---|---|
| LSTM | 9.09 minutes | 84.39 % |
| Naïve Bayes | 8.20 seconds | 55. 18 % |

## Conclusion

In this study, we collect a dataset of Indonesian-language tweets to detect and analyze hate speech and examine the performance of some of the features used. The dataset that we use is 3000 data. We label the tweets dataset into two classes: hate speech and neutral.

From the research conducted, it is concluded that the Word2vec feature is very influential on the LSTM method because the Word2vec feature is proven to be very accurate in processing data and can increase the accuracy of the LSTM method. The results also show that the Long Short-Term Memory method has an accuracy of 84.39% better than the Naïve Bayes method which has an accuracy of 55.18% and also the best accuracy for each parameter is epochs 50 with sigmoid activation function which has accuracy value 84.39%.

We compares the Naive Bayes Method and the LSTM Method because the Naive Bayes Method has been popular for decades, while the LSTM Methods started to find applications during the last decade due to their need for high computing resources and also most of the time trained on dedicated GPU's (which compute much more faster than CPU).

Scientific impact of this research is that can reduce the spread of hate speech on social media especially on Twitter, reduce the impact of black campaigns in politics, increase peace and security in the general public and can't be denied to increase prosperity in Indonesia.

For future work on this case, we suggest that the next study can use larger data in case of detecting hate speech and also in subsequent studies to combine the Long Short-Term Memory method with the Convultional Neural Network method for sentiment analysis classification.

## BIBLIOGRAFI

Alfina, I., Sigmawaty, D., Nurhidayati, F., & Hidayanto, A. N. (2017). Utilizing hashtags for sentiment analysis of tweets in the political domain. *Proceedings of the 9th International Conference on Machine Learning and Computing*, 43–47. Google Scholar

Alshalan, R., & Al-Khalifa, H. (2020). A deep learning approach for automatic hate speech detection in the saudi twittersphere. *Applied Sciences*, *10*(23), 8614. Google Scholar

Artur, M. (2021). Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features. *Procedia Computer Science*, *190*, 564–570. Google Scholar

Burnap, P., & Williams, M. L. (2014). *Hate speech, machine classification and statistical modelling of information flows on Twitter: Interpretation and communication for policy decision making*. Google Scholar

Mutanga, R., Naicker, N., & Olugbara, O. O. (2020). Hate speech detection in twitter using transformer methods. *International Journal of Advanced Computer Science and Applications*, *11*(01). Google Scholar

S. Biere. (2018). *"Hate Speech Detection Using Natural Language Processing Techniques."* Vrije Univ. Amsterdam. Google Scholar

Schmidt, A., & Wiegand, M. (2019). A survey on hate speech detection using natural language processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017, Valencia, Spain*, 1–10. Association for Computational Linguistics. Google Scholar

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). *â€ œClassification of sentiment reviews using n-gram machine learning approach, â€ Expert Syst*. Appl. Google Scholar

Turgut, Z., Aydin, G. Z. G., & Sertbas, A. (2016). Indoor localization techniques for smart building environment. *Procedia Computer Science*, *83*, 1176–1181. Google Scholar

V. Pathak, M. Joshi, P. Joshi, M. Mundada, and T. J. (2020). "Using machine learning for detection of hate speech and offensive code-mixed social media text,." *CEUR Workshop Proc*, *2826*, 351–361. Google Scholar

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. *Proceedings of the NAACL Student Research Workshop*, 88–93. Google Scholar

Zhang, C., Beetz, J., & de Vries, B. (2018). BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data. *Semantic Web*, *9*(6), 829–855. Google Scholar