Hate speech detection on Indonesian text using word embedding method-global vector

Mardhiya Hayaty¹, Arif Dwi Laksito¹, Sumarni Adi²
¹Department of Informatics, Faculty of Computer Science, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
²Department of Computer Science, Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo, Japan

Article Info

Article history:
Received Aug 30, 2022
Revised Jan 25, 2023
Accepted Mar 10, 2023

Keywords:
Abusive
Glove
Hate speech
Long short-term memory
Word embedding

ABSTRACT

Hate speech is defined as communication directed toward a specific individual or group that involves hatred or anger and a language with solid arguments leading to someone's opinion can cause social conflict. It has a lot of potential for individuals to communicate their thoughts on an online platform because the number of Internet users globally, including in Indonesia, is continually rising. This study aims to observe the impact of pre-trained global vector (GloVe) word embedding on accuracy in the classification of hate speech and non-hate speech. The use of pre-trained GloVe (Indonesian text) and single and multi-layer long short-term memory (LSTM) classifiers has performance that is resistant to overfitting compared to pre-trainable embedding for hate-speech detection. The accuracy value is 81.5% on a single layer and 80.9% on a double-layer LSTM. The following job is to provide pre-trained with formal and non-formal language corpus; pre-processing to overcome non-formal words is very challenging.

This is an open access article under the CC BY-SA license.

1. INTRODUCTION

Hate speech is defined as communication directed toward a specific individual or group that involves hatred or anger [1], and a language [2] with solid arguments leading to someone's opinion can cause social conflict. According to Indonesia's legal system, spreading hate speech on internet platforms is against the Information and Electronic Transactions Act, article 28 paragraph (2) “Every person who knowingly and without the right to disseminate information and aims is to incite hatred or hostility towards specific individuals and groups of people based on ethnicity, religion, race, and between groups”. Nowadays, Hate speech is not only done face-to-face but also through online communication [3]. There are numerous reasons for this, but one of the biggest is that people are so dependent on the internet and social media [4] that provocation is simple to disseminate and can lead someone to act illegally.

The Indonesian Internet Service User Association performed a study in 2022 to measure the number of internet users [5]. Identifying hate speech on social media is difficult because of harsh terms in regional languages. Everyone interprets sentences' meanings differently, and the same is true of hate speech and abusive language. Even if a term might have the same meaning and have components of both rudeness and hatred to different levels, such as rude, extremely rude, and truly hate, some people can perceive it as normal or simply a joke [6]. Terms that compare people to animals are frequently used in hate speech [7]. Examples include the words dogs, monkeys, "munyuk," pigs, boars, and others that have the same connotation.
In recent years, there has been an increase in cases of hate speech almost all over the world. Researchers have carried out many studies to detect hate speech to reduce this number. Several methodological approaches have been carried out [4] and the language factor as each region or country's cultural background makes it a challenge. Hate speech is a form of public opinion on the dissatisfaction with social phenomena that occur. Prior work [8] conducted an in-depth study of classification techniques to analyze these opinions. Hate speech detection in Indonesian text has been carried out [9] using a machine learning approach with the support vector machine (SVM). In addition, the ensemble method on several classification algorithms has been carried out to improve the classifier's performance; However, it does not produce significant performance; According to a study [10] an imbalanced dataset had an F1 measure of 79.8% while a balanced dataset had an F1 value of 84.7%.

Overcoming the weakness of the machine learning approach, research [11] proposes a deep neural network structure that functions as a feature extractor. It is very effective in capturing the semantics of hate speech and has increased the accuracy by 5%. The ensemble technique is not only used in machine learning but also in deep learning to improve the classification performance of hate speech and no-hate speech [12].

Hate speech classification performance in machine learning cannot be separated from the labeling process, which is quite laborious and time-consuming. One of the most important factors in supervised learning is the role of the annotator, but this role has the potential to racially bias the dialect, which is the concern of researchers [13] to suppress racial bias in African American English dialect. The problem of racism by annotators is carried out [14] by comparing the effect of the knowledge of an expert annotator with an amateur annotator on the classification model; the result is that amateur annotators are more likely to label hate tests. Some researchers work with different languages for hate speech classification, namely Arabic [15], Dravidian [16], and Russian [17] languages, and other works [18] detect hate speech with a combination of English-Hindi languages.

Research [6] employed multi-label labeling based on target, category, and level. In this work, multi-label hate-speech detection made use of a variety of feature extraction techniques. In order to get around this, the word embedding approach is utilized to explore word meanings using a word vector [19]. By utilizing Word2Vec, the word embedding approach was able to improve classification performance from 77.36% to 87.51%. Deep hybrid learning recurrent neural network (RNN)-LSTM and data balance on the dataset were used in the study [20] to identify abusive comments, which improved the F1 score. With the use of word embedding FastText, variations of BERT, xlm-Roberta, and distil-BERT, the work [21] transfer learning strategy to identify hate speech has an excellent F1 score. Additionally, the study [22] made use of Bert model trained on a big Spanish corpus (BETO), cross-lingual language model (XLM), and Bidirectional Encoder representations from transformers (BERT) which had already been trained. Utilizing GloVe and fastText with an LSTM classifier, pre-trained word embedding is also used for sentiment analysis [23].

This study aims to observe the impact of pre-trained GloVe word embedding on accuracy in the classification of hate speech and non-hate speech. The accuracy is an indicator of the study's success. In section 2, a number of experimental scenarios will be described. Section 3 discusses an experiment's outcomes, while Section 4 concludes and discusses further research.

2. METHOD
2.1. Framework

In this paper, a framework for LSTM-based hate speech detection and word embedding-based word vector construction is proposed. An Indonesian text hates speech detection framework is shown in Figure 1. Global vector (GloVe) is the word embedding technique employed and F-measure is being used as our evaluation matrix.

2.2. Experimental scenarios

This study aims to observe the impact of using GloVe word embedding on accuracy in classifying hate speech and non-hate speech. Pre-trained GloVe is a 100-dimensional Indonesian Wikipedia corpus. The splitting dataset includes 80% training and 20% testing data, while data validation consists of 20% training data. As a classifier method, the long short-term memory (LSTM) is employed. There are three scenarios in this study: a trainable embedding layer (without pre-training), a pre-trained GloVe, and a pre-trained GloVe-trainable embedding layer. Table 1 depicts the number of units, hyper-parameter values, and all scenarios used by single and multi-layer LSTM.

2.3. Dataset

There is 7,608 non-hate speech (non-HS) tweets and 5,561 hate speech (HS) tweets in the dataset, which contains 13,169 tweets [6]. When the imbalance ratio (IR) [24] is calculated and the categorization of
data is balanced (IR<9), the IR is 1.36. Or to put it another way, this dataset is balanced and does not require any balancing techniques.

![Proposed framework for hate speech detection](image)

**Table 1. Experimental scenarios**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Model</th>
<th>Hyper-parameter</th>
<th>Number of LSTM units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Trainable embedding layer</td>
<td>LR 0.001, Epoch 200, Dropout 0.3</td>
<td>8 Layer 1 &amp; 2 = 8</td>
</tr>
<tr>
<td>2</td>
<td>Pre-trained GloVe</td>
<td>LR 0.001, Epoch 200, Dropout 0.3</td>
<td>8 Layer 1 &amp; 2 = 8</td>
</tr>
<tr>
<td>3</td>
<td>Pre-trained GloVe+trainable embedding layer</td>
<td>LR 0.001, Epoch 200, Dropout 0.3</td>
<td>8 Layer 1 &amp; 2 = 8</td>
</tr>
</tbody>
</table>

2.4. Pre-processing

Case folding, data cleansing, stopword elimination, tokenizing, and stemming are all parts of pre-processing. To complete such processes, the author employed nltk and literary stemmer modules. The cleaning process removes both punctuation and numeric characters (@[A-Za-z0-9] +[^0-9A-Za-z \t]). Stopword removal removes words that often appear but have no meaning in the sentence. To overcome slang words, the word normalization procedure employs a slang dictionary [6]. Tokenization divides words, whereas stemming turns them into basic words by removing all affixes. There are 30,557 unique words in the dataset visualized as a consequence of tokenization; each tweet has an average word count of 11.2, a standard deviation of tokens of 7.04, and a maximum word length of 40.

2.5. Word embedding

The computer process only understands numbers, therefore word embedding converts words into a numeric vector. Word embedding is a method of turning words into a vector or array that is used to look for word relationships and meanings based on how close the distances between the vectors are to one another. Word proximity is described by word embedding, although not all words have the same meaning.

Words are points that are located in a certain location, hence the way word embedding works is to train continually on the vector area. Through various computations, these points may grow farther apart or closer dependent on other factors. Iteration until the points are unable to move any further; at the conclusion of iteration, neighboring words have similar meanings or are near in context. Figure 2 shows a straightforward instance of word embedding. The illustrates the Figure 2 mapping to the word embeddings, and the word “good” reflects close to “great” and far from “bad”. Word embeddings can improve the accuracy of classification models because they provide information and vector representation in the training data based on proximity to other words.

2.6. Global vector (GloVe)

Global vector (GloVe) is an unsupervised learning method of word representation to produce word embeddings-the gloVe method developed by Stanford University. GloVe handles word similarity, analogy, and named entity recognition [25]. Figure 3 depicts co-occurrence probabilities matrix. The GloVe algorithm uses probability theory, entering the probability of word occurrences in a window to obtain word semantics based on the co-occurrence matrix.

\[
P_{ik} P_{jk} \text{ where } P_{ik} = \frac{X_{ik}}{X_i}
\]
Hate speech detection on Indonesian text using word embedding method - global vector (Mardhiya Hayaty)

Figure 2. Word embedding illustration

<table>
<thead>
<tr>
<th>Probability and Ratio</th>
<th>$k = \text{solid}$</th>
<th>$k = \text{gas}$</th>
<th>$k = \text{water}$</th>
<th>$k = \text{fashion}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(k</td>
<td>\text{ice})$</td>
<td>$1.9 \times 10^{-4}$</td>
<td>$6.6 \times 10^{-5}$</td>
<td>$3.0 \times 10^{-3}$</td>
</tr>
<tr>
<td>$P(k</td>
<td>\text{steam})$</td>
<td>$2.2 \times 10^{-5}$</td>
<td>$7.8 \times 10^{-4}$</td>
<td>$2.2 \times 10^{-3}$</td>
</tr>
<tr>
<td>$P(k</td>
<td>\text{ice})/P(k</td>
<td>\text{steam})$</td>
<td>$8.9$</td>
<td>$8.5 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

Figure 3. Co-occurrence probabilities

$P_{ik}$ shows the probability of the occurrence of words $i$ and $k$ simultaneously, by dividing the number of times $i$ and $k$ appear together $X_{ik}$ to the total number of words $i$ appearing in the corpus $X_i$. Word in $k$ is “probe word”. As an example of two words, namely “ice” and “steam”.

Examples,

$k = \text{Solid}$ is very similar to ice but not similar to steam, then the probability of $P_{i/k}$ will be very high ($>1$)

$k = \text{Gas}$ is very similar to steam but not similar to ice, then the probability of $P_{i/k}$ will be very low ($<1$)

$k = \text{Water}$ is very similar to ice and steam or $k = \text{fashion}$ not similar to ice and steam, then the probability of $P_{i/k}$ will approach $1$

Therefore, in order to merge $P_{i/k}$ into the word vector count, global statistics are required while learning word vectors. Word vector is high dimension vectors, and $P_{i/k}$ is scalar. There are three-word entity ($i$, $j$, and $k$), and GloVe can provide a vector relationship between these three entities. Assume that a function $F$ represents the word vector of $i$, $j$ and $k$, which gives the ratio output in the following equation.

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$ (2)

Where,

$w, u$: separator between two embedding layers

$w^*$: transpose from $w$

$x$: co-occurrence matrix

$bw, bu$: bias $w$ and $u$

$P$: word probability

In (2) has two embedded embedding layers ($w$ and $u$). These two layers work equally and differ only in their random initialization. Having these two layers can help the model to reduce overfitting.

2.7. LSTM

A variation of the sequence model for recurrent neural networks is the LSTM. The vanishing gradient is a shortcoming of RNN that is solved by LSTM architecture. LSTM cells are used to store past data. The input gate, forget gate, and output gate are the three gates of LSTM cells, which are used to read, store, and update prior information [26]. LSTM unit is shown in Figure 4.

Where,

$X_t$ = Input vector at the time $t$

$h_{t-1}$ = Previous Hidden state.

$C_{t-1}$ = Previous Memory state.

$h_t$ = Current Hidden state.

$C_t$ = Current Memory state.

$[x]$ = Multiplication operation.

$[+]$ = Addition operation.
Starting from *forget gate*, data $x_t$ is data input (vector input $x$ in *timestep* $t$), and $h_{t-1}$ is *hidden state vector* in previous *timestep* $t-1$). The input gate component processes information and forms a new candidate using the activation function tanh, then updates the cell state value $C_{t-1}$ to a new cell state $C_t$. Finally, the output gate component runs sigmoid and produces an output value in the hidden state ($h_t$). In this work, we implement a single-layer and double-layer LSTM, with 8 LSTM units, maximum input is 31 denote $X$, and the value of $t$ is 1…31. The LSTM architecture we employed is illustrated in Figure 5. The image in Figure 5(a) is LSTM architecture for a single-layer while Figure 5(b) is for a double-layer that has 8 unit of memory for each layer.

![Figure 4. The LSTM unit](image)

![Figure 5. LSTM architecture (a) Single-layer and (b) Double-layer](image)

3. **RESULTS AND DISCUSSION**

Wikipedia provides the Big Corpus in Indonesian, and the glove trains into an embedding vector, yielding 378,000-word vectors using 100 dimensions. Pre-trained input embedding layer is the end output. The vocabulary hates speech dataset has a size of about 24,000, and the maximum pad-sequence length is 31. The long short term memory (LSTM) method is used to classify hate-speech or non-hate-speech. Word embedding as input to the LSTM to generate a classification model. Word embedding layer in scenario 1 without pre-trained GloVe, scenario 2 using pre-trained GloVe, while the last scenario combines scenario one and scenario 2. The experiment’s findings are shown in Table 2.
The pre-trained model has fewer parameters than other models, which has an impact on the training process time, which is only 1.020s, or twice as fast as other models. The graphs of the modeling results are shown in Figures 6 and 7. In comparison to other models, the Pre-trained Glove+trainable embedding layer model (scenario 3) achieves the greatest accuracy of 82.6 on double-layer LSTM. Figure 7(c) is the graphic of accuracy scenario 3. However, in cases scenarios 1 and 3 there is a large amount of overfitting. Figures 6(a), 6(c), 7(a) and 7(c) is graphic accuracy that overfitting appears in both single-layer and double-layer LSTM.

Figure 6. Training accuracy of single layer LSTM (a) Trainabled embedding, (b) Pre-trained GloVe embedding and (c) Pre-trained GloVe+trainable embedding.
The best scenario is scenario 2 because there is no overfitting. The GloVe has two layers of embedding inserted (see (2)), so the pre-trained Glove model is more resistant to overfitting for single-layer and double-layer LSTM. Figure 6(b) and 7(b) results from scenario 2. Besides being more resistant to overfitting, scenario 2 is faster than the other scenarios, especially in a single layer. The Pre-Trained GloVe has decent performance and merits consideration for speed and handled of overfitting because the accuracy of other models is not too significant.

Figure 7. Training accuracy of double-layer LSTM (a) Trainabled embedding, (b) Pre-Trained GloVe embedding and (c) Pre-trained GloVe+trainable embedding
In this study, we analyze the prediction errors in the classification of hate speech, as shown in Table 3. Corpus Wikipedia generally uses formal language and common words (no slang); this is in contrast to crude language, which tends to use informal language (the use of regional languages) and irregular word or sentence structures (slang, hyperbole). In the informal type (see Table 3, highlight word), the word “anjir/dog” is slang from the formal word “anjing/dog”, while “lo” or “lu” comes from the term “you”, the model is unable to classify as hate speech.

In Indonesian culture, equating humans with animals is a very terse statement. Pre-trained GloVe (see Table 4) interprets the words “anjing/dog”, “babi/frog”, and “monyet/monkey” in the context of animal names, not in the context of abusive or hateful statements. So, the category of sentence classification results is no-hate speech. Indonesia is an archipelagic country that has a diversity of local languages, generally using that language in everyday conversation, including hate speech. The first sentence (see Table 3, local languages) is Sundanese, and the second sentence is Javanese. The word "pisan" means "very" and "caileh" as a form of joke. The Javanese language "cangkemmu dewek" means "your own mouth", including rude words in the Javanese tribe. The model is not able to classify correctly. This study improved accuracy in comparison to research [7], which led to an accuracy of 77.36% utilizing the same dataset in this experiment.

### Table 2. Accuracy

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Model</th>
<th>Trainable Parameters</th>
<th>Time (s)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Single Layer LSTM</td>
<td>Double Layer LSTM</td>
<td>Single Layer LSTM</td>
</tr>
<tr>
<td>1</td>
<td>Trainable embedding layer</td>
<td>2,418,949</td>
<td>2,419,493</td>
<td>1,920</td>
</tr>
<tr>
<td>2</td>
<td>Pre-trained GloVe</td>
<td>3,649</td>
<td>589</td>
<td>1,020</td>
</tr>
<tr>
<td>3</td>
<td>Pre-trained GloVe+trainable embedding layer</td>
<td>2,418,949</td>
<td>2,419,493</td>
<td>2,100</td>
</tr>
</tbody>
</table>

### Table 3. Classification error analysis

<table>
<thead>
<tr>
<th>Type</th>
<th>No</th>
<th>Text</th>
<th>Label</th>
<th>Predict</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formal</td>
<td>1</td>
<td>gubernur dki jakarta &lt;subject&gt; kehormatan bertemu sholat jumat presiden &lt;subject&gt;</td>
<td>No-hate speech</td>
<td>No-hate speech</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>governor of DKI-Jakarta &lt;subject&gt; the honour of meeting the president’s Friday prayer &lt;subject&gt;</td>
<td>Hate speech</td>
<td>Hate speech</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>gampangan banget siborang dipuji dikt aja udah kaya ngasih kehormatannya aja haha tolo It’s so easy for people to be praised even a little bit like giving honour to it laugh emotion... stupid</td>
<td>Hate speech</td>
<td>Hate speech</td>
</tr>
<tr>
<td>Informal</td>
<td>1</td>
<td>yah dah bobo si &lt;subject&gt; trus kup cher anjir lo najis Yes!...already sleeping &lt;subject&gt;, so what? You’re a dog, and you’re profane</td>
<td>Hate speech</td>
<td>No-hate speech</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>cupu anjir lu you’re not very pleasant, you’re a dog</td>
<td>Hate speech</td>
<td>No-hate speech</td>
</tr>
<tr>
<td>Animals</td>
<td>1</td>
<td>&lt;subject&gt; paham si babi &lt;subject&gt; klo bicara santun bangsat taak anjing setan lo dgn ini orgasm yg familiar tuntuna &lt;subject&gt; comprehend the pig &lt;subject&gt;? if you talk politely, bitch, dog shit, you’re a devil</td>
<td>Hate speech</td>
<td>No-hate speech</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>si &lt;subject&gt; pantasnya dinobatkan sbgai babi ngepet &lt;subject&gt; deserves to be crowned as a sucking pig</td>
<td>Hate speech</td>
<td>No-hate speech</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>coba kalo &lt;subject&gt; yg ngomong gtu bani kampret nurka &lt;subject&gt; alumni kampret monyet If &lt;subject&gt; says that &lt;subject&gt; is shucks, angry &lt;subject&gt; is alumni, a moron, a monkey</td>
<td>Hate speech</td>
<td>No-hate speech</td>
</tr>
<tr>
<td>Local languages</td>
<td>1</td>
<td>jipepet ngeselin pisan anaknya kdang saka bener ama monyet cowonya seganbreng caileh That woman is annoying, sometimes she likes monkeys, has a lot of guys... caileh</td>
<td>No-hate speech</td>
<td>Hate speech</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>wkwwkwkwk cebong dungu yg ngomong cebong onta cangkemmu dewek wkwwkwk stupid tadpole who talks about frogs, camels, cangkemmu</td>
<td>Hate speech</td>
<td>No-hate speech</td>
</tr>
</tbody>
</table>

### Table 4. Closest embeddings

<table>
<thead>
<tr>
<th>Word</th>
<th>Top ten closest embeddings</th>
<th>Top ten closest embeddings (in English)</th>
</tr>
</thead>
<tbody>
<tr>
<td>anjing</td>
<td>[anjing', 'peliharaan', 'seekor', 'kucing', 'binatang', 'rusa', 'beruang', 'pembunuh', 'kelinci', 'hewan']</td>
<td>['dog', 'pet', 'u', 'cat', 'animal', 'deer', 'bear', 'hunter', 'rabbit', 'animal']</td>
</tr>
<tr>
<td>babi</td>
<td>[babi', 'daging', 'sapi', 'kambing', 'ayam', 'domba', 'kelinci', 'kerbau', 'peliharaan', 'panggang']</td>
<td>['pork', 'meat', 'beef', 'goat', 'chicken', 'lamb', 'calidonia', 'buffalo', 'rabbit', 'roast']</td>
</tr>
<tr>
<td>monyet</td>
<td>['monyet', 'kera', 'vervet', 'tikus', 'wolai', 'peliharaan', 'hesus', 'kelinci', 'ngipiri', 'andarbeni']</td>
<td>['monkey', 'ape', 'vervet', 'rat', 'wolai', 'pet', 'hesus', 'rabbit', 'ngipiri', 'andarbeni']</td>
</tr>
</tbody>
</table>
4. CONCLUSION

In comparison to pre-trainable embedding, the usage of pre-trained GloVe with single-layer and double-layer LSTM classifiers offers performance that is resistant to overfitting. On a single layer, the accuracy value is 81.5%, while on a double-layer LSTM, it is 89.9%. The next task is to supply pre-trained language corpora in both formal and informal dialects (regional languages). Moreover, pre-processing to overcome non-formal words will be very challenging for further research.

ACKNOWLEDGEMENTS

The authors received financial support from the research grant No.044/Lemlit/AMIKOM/VII/2020 from Directorate Research Universitas AMIKOM Yogyakarta.

REFERENCES


Hate speech detection on Indonesian text using word embedding method-global vector (Mardhiya Hayaty)

BIOGRAPHIES OF AUTHORS

Mardhiya Hayaty is currently working as Assistant Professor, at Informatics Department in Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia. Her research interest is natural language processing, sentiment analysis, and automatic text summarization. She can be contacted at email: mardhiya_hayati@amikom.ac.id.

Arif Dwi Laksito holds a Bachelor of Computer Science (S. Kom) and a Master of Engineering (M. Kom) in Computer Engineering in 2006 and 2013, respectively. He is currently a researcher in Natural Language Processing and a Lecturer at the Faculty of Computer Science Universitas Amikom Yogyakarta. His research areas of interest include the Recommender system, Sentiment analysis, and Named Entity Recognition. He can be contacted at email: arif.laksito@amikom.ac.id.

Sumarni Adi is an Indonesian studying PhD at Tokyo Metropolitan University in Tokyo, Japan. Her major is Computer Science, focus on Machine Learning and Deep Learning. She offered MEXT scholarship for her PhD studies. She is an Assistant Professor at Information Systems Department in Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia as well. She is an active member of IEEE, she organized technical events for IEEE International Conference in the IEEE Indonesia Section. She can be contacted at email: sumarni-adi@ed.tmu.ac.jp.