ISBN: 978-981-18-1791-5
2021 the 11th International Workshop on Computer Science and Engineering (WCSE 2021)
doi: 10.18178/wcse.2021.06.029

Long Short-Term Memory for Hate Speech and Abusive Language Detection on Indonesian Youtube Comment Section

Calvin Erico Rudy Salim [1], Derwin Suhartono [2 +]

[1] Computer Science Department, BINUS Graduate Program, Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

[2] Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Abstract. Hate speech is one of the most challenging problem internet is facing today. With increasing numbers of users online, hate speech also rise and takes time to be classified manually particularly in languages other than English. This research examines hate speech detection problem in form of Bahasa Indonesia. Millions of comments and text posts are added to various social media and discussion platforms. Manual classification in all of the internet as hate speech and offensive language is a near impossible and time-consuming task. This research uses Long Short-Term Memory (LSTM) and Bidirectional Long Short Term Memory (Bi-LSTM) for the method of classifying hate speech and abusive language. The final accuracy is 88,44% by using 200 neurons with Bi-LSTM method. Most common challenges are different languages, out of vocabulary words, long range dependencies, and sarcasm.

Keywords: hate speech, machine learning, natural language processing

1. Introduction

The social media in internet is often exploited and abused to spread content that can harass or harm someone. An important and difficult to understand form of such language is hateful speech, a content that expresses hatred towards a group in society. Hateful speech has become a major problem for every type of online platform, from the comment section of news websites to real-time chat in online gaming. Hate speech also has been increasing and new users are becoming hateful at a faster rate [1] . There are at least 3.325 hate speech case in Indonesia that has been handled by police in 2017 and the numbers are still growing especially in presidential election [2] . Such content can alienate users and can also support radicalization and trigger violence [3] . Hate speech and abusive language can cause physical and psychological suffering. Most cases of hate speech and harsh language occur in minorities and indigenous people [4] . Because social media in internet offers such level of anonymity, for example we can create email address easily with fake names and fake address, hate speech can occur easily and could be so hard getting caught [5] .

Other platforms recognize that hateful content poses practical, ethical problems and many others, including Twitter, Facebook, Reddit, Youtube and game companies such as Riot Games, have tried to prevent this, by changing their platforms or policies. However, reliable solutions for online hate speech are lacking [6] . Currently, the platform relies mostly on users to report objectionable content. This requires a labor intensive review by platform staff and can also completely overlook any hateful or harmful speech or other underreported dangers. With a high volume of content being generated on large platforms, accurate automated methods may be a useful measure to reduce the effects of hate speech. Unfortunately, every user who engages in online social media will always have the risk of being targeted or harassed through hate speech and offensive language, expressing hatred in the form of racism or sexism, which is likely to have an impact on the online experience and the community in general [7] . However, with the development of technology, humans can quickly prevent the problem of hate speech, one of which is Artificial Intelligence (AI). AI is an artificial intelligence that is demonstrated by a machine or computer, so it can solve complex problems and complex mathematical calculations quickly. More specifically for the problem of hate speech, Natural Language Processing (NLP) will be used, which is the sub-field of linguistics and AI that deals with

+ Corresponding author. Tel.: +62215345830; fax: +62215300244.
E-mail address: dsuhartono@binus.edu.

interactions between computers and human (natural) language, in particular on how to program computers to process and analyze large amounts of natural language data.

Youtube has 2 billion active users every month in 2019. It is estimated that there are 500 hours of video uploaded every minute and each user spends an average of 11 minutes 24 seconds per day on Youtube [8] . According to these sources 6 out of 10 people prefer online video platforms over television. From these data, it cannot be denied that Youtube has a lot of users. However, because so many users also comment on videos, Youtube can also become a hotbed of hate speech and offensive language. Hate comments written by people can be seen easily and can hurt someone's feelings so that they can make the person depressed, angry, trigger a scene, and even commit suicide [9] . Therefore, this study focuses on preventing hate speech through the comment column of the number one online video platform in the world. Twitter on the other hand, is the most used dataset to train and test computational models especially in text classifications. Other research [10] and [11] use Twitter dataset because of its simplicity and few limit characters.

The difference between hate speech and abusive language is that hate speech is a term that hits certain communities or individuals who make them suffer, while the opposition doesn't care [12] . While offensive language is speech that contains harsh words / phrases that are conveyed to the interlocutor (individual or group), both orally and in writing [13] . In this study, we will focus on using comments from Youtube as a test dataset and Twitter as a training dataset with the method to be used is Long Short-Term Memory (LSTM) which is the architecture of the Recurrent Neural Network (RNN) to detect hate speech and offensive language. Previous research on hate speech detection and offensive language has identified this problem but many studies still tend to use the Support Vector Machine (SVM), Random Forest Decision Tree (RFDT), and Naive Bayes.

2. Related Works

Research on the detection of hate speech and offensive language has been done a lot, but most of these studies use English. As done [14] in their research, they made a classification of hate speech comments with Convolution Neural Network (CNN) where the dataset used came from Twitter users and comments on Wikipedia provided by Google Jigsaw on Kaggle's Toxic Comment Classification Challenge. This study is the first to apply and compare various classification methods with a new public multi-label dataset containing over 200,000 user comments. The accuracy of the F1-score in the Wikipedia dataset using CNN combined with the FastText word embedding reaches an average F1-score of 0.776. Meanwhile, CNN combined with the word embedding GloVe reached an average F1-score of 0.748. The Twitter dataset using the CNN word embedding method FastText reached an average F1-score of 0.775, and the CNN word embedding method GloVe reached an average F1-score of 0.769.

The use of the CNN method was also carried out by [15] where the research used an annotated Twitter dataset. The dataset contains approximately 16,000 tweets of which 3,383 were labeled sexist, 1,972 were labeled racist and the rest were marked as neither sexist nor racist. The study performed 10-fold cross validation and calculated macro precission, recall and F1-score. The result is that with the CNN method and the GloVe word embedding method, the macro precssion gets 0.839, the recall gets 0.840 and the F1-score gets 0.839.

Apart from CNN, the methods that are often used are RNN and LSTM. Like research conducted by [7] using a dataset from Twitter of approximately 16,000 tweets containing 1,943 tweets labeled racism, 3,166 tweets labeled sexism and 10,889 tweets labeled neutral or those that are not sexism or racism. The distribution of the dataset was done randomly with 15% of the validation data and 85% of the training data. The training is made to 100 epochs in order to avoid overfitting, and to achieve stability in results, run each single classifier for 15 times and the output values are aggregated. The result is an F1-score of 0.9320 using the LSTM method.

In 2016, [16] presented Tweet2Vec, the creation of a new method for generating general purpose vector representations of tweets. Tweet2Vec uses a CNN-LSTM encoder-decoder model that operates at the character level to learn and generate vector representations of tweets. Apart from Tweet2Vec, Word2Vec was also made by [17] to be used as word embeddings. However, Tweet2Vec is used specifically for tweets

from Twitter. The research [16] aims to predict semantic equivalence through a binary yes / no assessment using a training dataset which contains around 18,000 tweets and a test dataset containing around 1,000 tweets where 35% of these pairs are paraphrased, and 65% non- paraphrase. As a result, using the Tweet2Vec method, the F1-score reached about 0.677, the recall reached 0.686 and precision 0.679.

From previous research, the accuracy is quite high, even close to the perfect value of 0.93. However, the research that has been done is still using the dataset from Twitter in English. This research focuses on the comments column on the Youtube platform to be used as test data and Twitter datasets for training data, where active users when compared to Twitter, Youtube has far more active users. There has been no specific research on the detection of hate speech and abusive language using Long Short-Term Memory on the Indonesian-language Youtube platform, which encourages this research to be carried out.

3. Proposed Methods

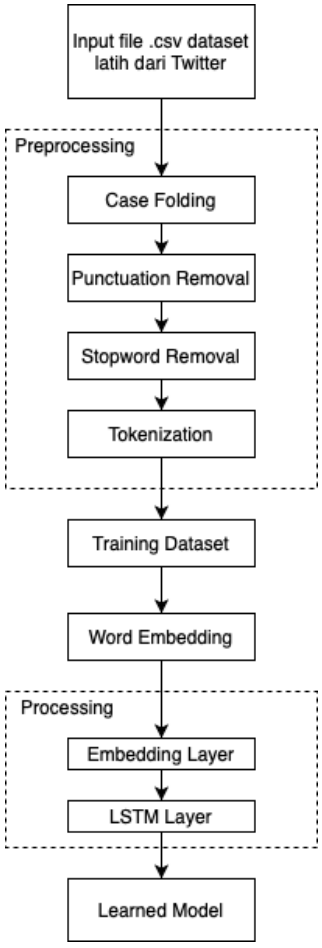

Fig. 1: General architecture system

A dataset is a collection of data or records that store the value in each variable [18] . The dataset for neural network training in this study will use a dataset from [19] which contains 13,169 tweets from Twitter. 7,608 of them were tweets that did not contain elements of hate speech and offensive language, and 5,561 tweets contained elements of hate speech and offensive language. The training dataset was taken using the method of crawling and scraping techniques using Twitter's Application Programming Interface (API) which is implemented using the Tweepy Library. This training dataset has gone through the annotation process by conducting discussions and consultations with linguistic experts by author in order to obtain valid annotation guidelines and high standard annotations.

This research method consists of several main stages. A file with the csv extension that stores data for model training that contains tweets labeled hate speech, abusive language, both, and none. Then the training dataset goes through a preprocessing process consisting of case folding, punctuation removal, stopword

removal, tokenization. The training data is then formatted from text into a 2D tensor of integer. Then the model training uses the LSTM algorithm and ends with testing the model using the test dataset containing hate speech comments and offensive language taken from the Youtube comments column.

3.1. General Architecture System

Starting from the stage of identifying the problem that occurred. In this first stage, it will be done by looking at Youtube and looking for hate speech that has occurred in several videos. This process is carried out so that we can find out if hate speech and harsh language raises how bad conditions are in the online video-based world.

After identifying the problem, a literature study [20] was carried out to become a research topic. In fact, very few research has been done to eliminate hate speech and harsh Bahasa Indonesia in the comment's column on Youtube. Therefore, this study will propose a model that can classify hate speech comments and offensive language based on the definition of hate speech with the Long Short-Term Memory technique. Then proceed to the preparation stage of the training dataset by collecting data for processing. The dataset to be used for training is a dataset that has been made public from research [19] . The dataset contains tweets from social media Twitter that have been labeled and annotated in consultation with linguistics experts with the labels hate speech, abusive language, both and none.

The next step is implementation, to train the model that has been created with the training dataset from Twitter to classify hate speech comments and offensive language by entering the test dataset that has been searched on the Youtube platform using the API from Youtube. After all stages have been carried out, it ends with data evaluation. Analyzing the data aims to find out if the proposed method is suitable for purpose or not. Then conclusions will be drawn from the results of the implementation of this research.

3.2. Preprocessing

Case folding changes the characters in a sentence into a lowercase which aims to equalize all types of characters contained in comments so as to facilitate the process of deleting certain unwanted characters or words in this study. At this stage punctuation marks (such as?!, / = + - \> <; "() {} [].: | And others) will be replaced with spaces. This deletion is done because punctuation is ignored during the training process so that by removing punctuation marks the training process will be simpler. The exception to the "." and ":" which will be handled in the URL removal process at a later stage. Stopwords are words that occur frequently and are usually ignored in processing. Stopword removal aims to reduce the number of words in a document which will later affect the speed and performance of NLP activities. Tokenization functions to separate each word, a series of numbers, and a series of numbers with letters that have certain meanings.

Table 1 . Preprocessing Data Example

Example	**English Translation**
Mereka kalo makan pedes :"wah ini pedesnya mantap INI" Gw kalo makan pedes :"njir pedes banget cok,minum WOI minum!!"	If they want to eat spicy :"Wow THIS spice is great" Me if want to eat spicy :" man so spicy oh my drink PLEASE drink!!"

Preprocessing	**Text**
Case folding	mereka kalo makan pedes :"wah ini pedesnya mantap ini" gw kalo makan pedes :"njir pedes banget cok,minum woi minum!!"
Punctuation removal	mereka kalo makan pedes wah ini pedesnya mantap ini gw kalo makan pedes njir pedes banget cok minum woi minum

Stopword removal	makan pedes pedesnya mantap gw makan pedes njir pedes banget cok minum woi minum
Tokenization	[makan] [pedes] [pedesnya] [mantap] [gw] [njir] [banget] [cok] [minum] [woi]

3.3. Data Split

At this stage, the data that has been obtained are divided into two groups, namely the trained data and the tested data. The training data is used to train the model so that it can classify hate speech or not, and the training data is used to test the model that has been made so that you can find out whether it is accurate or not. The data sharing will be 60% for training data, 20% for validation data and 20% for test data. Training data is data used to train a model that has been created. The goal is to train the Recurrent Neural Network algorithm that has been made to match the desired results. The type of learning is supervised learning, where the data to train the model has a label. Validation data is data taken from a dataset but separate from training data. Validation data aims to validate the model during the training process and provide information that can help adjust hyperparameters. Another goal is to make the model not overfitting or underfitting.

3.4. LSTM Implementation

This study will use LSTM as a method of testing the model that has been made. LSTM is the optimal architecture of the RNN for sequential tasks such as text on sentences. When building the model, the first step taken is determining the input shape and input size of the LSTM neural network, then determining how many hidden layers you want to use. After that, the function for flatten is entered in order to change the pooled feature map into one column which will be moved to the fully connected layer. Then, dense adds a fully connected layer to the neural network. The next step is to determine the loss and optimizer that will be used in the model. Each determination made does not have to use the same value, because fine-tuning is done so that the values used can be different in order to achieve the best results. Below shows figure of LSTM model from [21] .

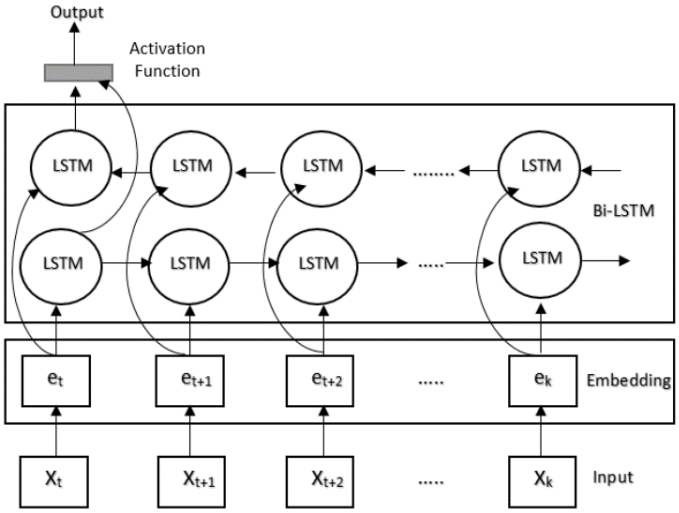

Fig. 2 : LSTM Model

In LSTM, there are internal mechanisms called gates that can regulate the flow of information. These gates can learn which order of data is important to keep and which to discard. The LSTM equation according to [22] is broken down into the following equation (1)

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

To update the cell state, there is an input gate. First, information from the previous hidden state and information from the current input will pass through a sigmoid function which will decide which value to update by changing the value between 0 and 1. 0 means unimportant and 1 means important. Then after that, the value is passed through the tanh function to update the value to between -1 and 1 to help set up the

LSTM network. After that, the yield of tanh and sigmoid is multiplied in order to decide which value should be stored. The gate input equation is broken down into equation (2)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

Furthermore, there are equations for the new candidates which are described as equation (3)

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

After having sufficient information, the next step is to update the old cell state, namely Ct-1, with a new value, namely Ct, as shown in Figure x. The cell state equation is described in equation (4)

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Finally, at the last stage there is an output gate. The output gate serves to determine the hidden state for the next timestamp.

4. Results and Discussion

The system's performance was measured with standard metrics for classification accuracy, namely Precision, Recall and F1-Score. F1-Score is the harmonic mean of precision and recall. The data is divided into four classes, namely, hate speech but not abusive language, abusive language but not hate speech, both hate speech and abusive language, not hate speech nor abusive language. The dataset that contains training tweets and testing comments from Youtube are in Bahasa Indonesia.

The best amount of neurons can be used for further parameter testing. In previous tests, the number of neurons has been determined based on test results that show the best value. In this study, the number of epochs tested was 1 to 10 while the number of neurons was 200 because it had the highest accuracy based on previous testing and learning rate value of 0.001. The first thing we test is using LSTM method, and we can see it from Table 2.

Table 2 . Result of Testing Twitter Dataset With LSTM Method

Epoch count	Time (minutes)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Loss
2	01:22	65.06	69.30	60.25	64.38	0.8973
4	02:40	72.48	76.46	68.99	72.48	0.8013
6	04:04	**75.21**	**77.50**	**73.14**	**75.21**	**0.7624**
8	05:22	73.08	74.14	71.76	72.91	0.9003
10	06:40	73.08	74.97	71.90	73.37	0.9664

As we can see from Table 2, the best outcome is from the sixth epoch count which has 75.62% of accuracy, 75.21% of F1-Score and although the loss almost get to 1, the lowest loss recorded is 0.7624. After completing this test, we continue to use the same hyperparameters but using a different method. This time we are using Bi-LSTM and the results are on Table 3.

Table 3 . Result of Testing Twitter Dataset With Bi-LSTM Method

Epoch count	Time (minutes)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Loss
2	00:24	**94.49**	**0.9493**	**0.9418**	**94.55**	**0.1752**
4	00:47	92.16	0.9258	0.9168	92.12	0.2768
6	01:10	90.70	0.9111	0.9058	90.84	0.3962
8	01:32	89.10	0.8934	0.8897	89.15	0.4944
10	01:54	88.44	0.8866	0.8832	88.49	0.6143

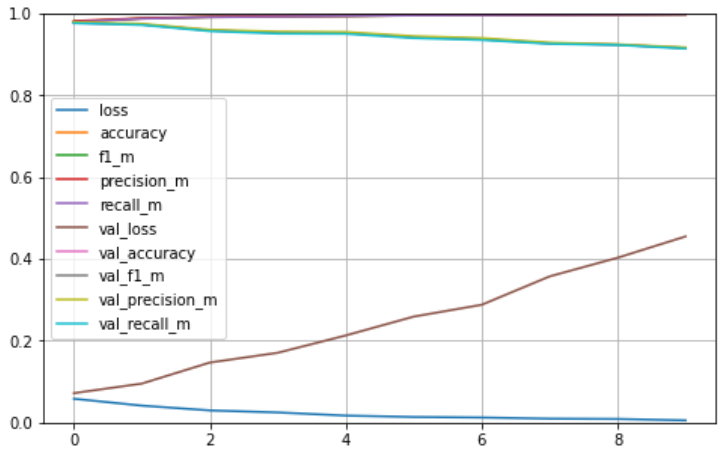

Fig. 3: Graph of Bi-LSTM Model

From the test results in Table 3, we can see that Bi-LSTM at the second epoch gets the highest accuracy value which is 94.49%. This shows that the use of bidirectional LSTM layer can increase the accuracy value in the model, and also using this method can decrease training time for the model. In Figure 3, there are 10 legend colors but only 5 are easily visible. This is due to the results are very similar, for example val_f1_m and val_accuracy have the score of 94.49% and 94.55% on the second epoch respectively.

We can see that Bi-LSTM performs better than LSTM because Bi-LSTM is able to predict the upcoming words due to its bidirectional behavior. It creates another layer of LSTM but going to the opposite direction, so that the output is known on every step combined with an activation function. Because it's Bi-LSTM performs better in our test, we then continue the test to implement our model to our own data from Youtube comments. Table 4 shows the result of the test.

Table 4 : Result of Predicting Youtube Comments

Description	**Total Comments**
Hate Speech but not Abusive Language	157
Abusive Language but not Hate Speech	152
Both Hate Speech and Abusive Language	62
Not Hate Speech nor Abusive Language	606
Total	977 comments

All these videos were taken from topics such as politics, comedy, gaming, sports and music videos randomly across Youtube. Each video we take 100 comments to be scraped and put in our own created dataset. From the results of Table 4, we can see that the most comments are not hate speech nor abusive language. Both hate speech and abusive language only got 62 detected by our model.

5. Conclusions

The use of Bidirectional method of Long Short Term Memory with 10 epoch, learning rate 0.001 and the number of neurons 200 on the hidden layer, generates an accuracy rate of 84,44% and 88,49% of F1-Score. In order to achieve this, the model needs to be fine tuned and have a clean dataset. Hate speech detection in Bahasa Indonesia is still very rare because it's not easy to find a clean dataset and there's not much database of stopwords or lemmatized words in Bahasa Indonesia.

For the future or next research, we suggest finding a clean dataset or create a dataset with sentences that's already been manually labeled. Preprocess data is also very important in order to achieve higher scores. Also, we suggest try using other method of machine learning such as using BERT or Convolutional Neural Network (CNN). Besides, we also recommend using Indonesian Part-of-Speech Tagger in detecting hate codes to determine the noun phrases as candidates for hate codes.

6. References

[1] Mathew, B., Illendula, A., Saha, P., Sarkar, S., Goyal, P., & Mukherjee, A. (2020). Hate begets hate: A temporal study of hate speech. Proceedings of the ACM on Human-Computer Interaction, 4(CSCW2), 1-24.

[2] Perwirawati, E., Simamora, P. R. T., & Sinaga, L. V. (2019). Pola Komunikasi Kelompok Agama Dalam Pencegahan Penyebaran Ujaran Kebencian Di Kecamatan Medan Polonia Pada Pemilihan Presiden Tahun 2019. Jurnal Darma Agung, 27(3), 1124-1134.

[3] Allan, J. (2013). The harm in hate speech. Constitutional Commentary, 29(1):59–80.

[4] Anam, M. Choirul & Muhammad Hafiz. (2015). "Surat Edaran Kapolri tentang Penanganan Ujaran Kebencian (hate speech) dalam Kerangka Hak Asasi Manusia" dalam Jurnal Keamanan Nasional. 1 (3) 41-364. http://www.jurnal.ubharajaya.ac.id/index.php/kamnas/article/viewFile/ 30/23. Diakses pada 20 Maret 2018.

[5] Mondal, M., Silva, L. A., & Benevenuto, F. (2017, July). A measurement study of hate speech in social media. In Proceedings of the 28th acm conference on hypertext and social media (pp. 85-94).

[6] Saleem, H. M., Dillon, K. P., Benesch, S., & Ruths, D. (2017). A Web of Hate: Tackling Hateful Speech in Online Social Spaces.

[7] Pitsilis, G., Ramampiaro, H., & Langseth, H. (2018). Effective hate-speech detection in Twitter data using recurrent neural networks. https://doi.org/10.1007/s10489-018-1242-y

[8] Clement, J. (2020). YouTube - Statistics & Facts. Retrieved from https://www.statista.com/topics/2019/youtube/

[9] Anagnostou, A., Mollas, I., & Tsoumakas, G. (2018). Hatebusters: A Web Application for Actively Reporting YouTube Hate Speech. In IJCAI (pp. 5796-5798).

[10] Bisht, A., Singh, A., Bhadauria, H. S., & Virmani, J. (2020). Detection of hate speech and offensive language in twitter data using lstm model. In Recent Trends in Image and Signal Processing in Computer Vision (pp. 243-264). Springer, Singapore.

[11] Huang, X., Xing, L., Dernoncourt, F., & Paul, M. J. (2020). Multilingual Twitter corpus and baselines for evaluating demographic bias in hate speech recognition. arXiv preprint arXiv:2002.10361.

[12] Anam, M. C., & Hafiz, M. (2015). Surat Edaran Kapolri Tentang Penanganan Ujaran Kebencian (Hate Speech) dalam Kerangka Hak Asasi Manusia. Jurnal Keamanan Nasional, 1(3), 341-364.

[13] Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016, April). Abusive language detection in online user content. In Proceedings of the 25th international conference on world wide web (pp. 145-153).

[14] Aken, B., Risch, J., Krestel, R., & Löser, A. (2017). Challenges for Toxic Comment Classification: An In-Depth Error Analysis. https://doi.org/10.18653/v1/W18-5105

[15] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep Learning for Hate Speech Detection in Tweet. http://dx.doi.org/10.1145/3041021.3054223

[16] Vosoughi, S., Vijayaraghavan, P., & Roy, D. (2016). Tweet2Vec. Proceedings Of The 39Th International ACM SIGIR Conference On Research And Development In Information Retrieval - SIGIR '16. https://doi.org/10.1145/2911451.2914762

[17] Mikolov, T., Sutskever, I., Chen, K., Corrado, S., G., and Dean J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

[18] Snijders, C., Matzat, U., Reips, U.-D. (2012). 'Big Data': Big gaps of knowledge in the field of Internet. International Journal of Internet Science. 7: 1–5.

[19] Ibrohim, O. M. & Budi, I. (2019). Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter. In ALW3: 3rd Workshop on Abusive Language Online, 46-57.

[20] Salim, C. E. R., & Suhartono, D. (2020). A Systematic Literature Review of Different Machine Learning Methods on Hate Speech Detection. JOIV: International Journal on Informatics Visualization, 4(4), 213-218.

[21] Isnain, A. R., Sihabuddin, A., & Suyanto, Y. (2020). Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection. IJCCS (Indonesian Journal of Computing and Cybernetics Systems), 14(2), 169-178.

[22] Hochreiter S. & Schmidhuber J. (1997). Long short-term memory. Neural Computation, 9:1681–1726.