Detection of Hate Speech by Employing Support Vector Machine with Word2Vec Model

Nina Sevani
Informatics Department
Krida Wacana Christian University Jakarta,
Indonesia
nina.sevani@ukrida.ac.id

Iwan A. Soenandi
Industrial Engineering
Krida Wacana Christian University
Jakarta, Indonesia
iwan.as@ukrida.ac.id

Adianto
Informatics Department
Krida Wacana Christian University Jakarta,
Indonesia
adianto.2016tin010@civitas.ukrida.ac.id

Jeremy Wijaya
Informatics Department
Krida Wacana Christian University Jakarta,
Indonesia
jeremy.2017tin005@civitas.ukrida.ac.id

2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE) | 978-1-6654-3232-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICEEIE52663.2021.9616721

***Abstract* – Social media can be seen as one prominent assimilation of technology into human life interaction. Its presence is now likely inseparable to us and its usage, whether individually or communally, evidently has been impactful by means of news spread– both positive and negative ones. As a highlight, Indonesia records more than 3,640 hate speech cases from 2018 to this day. This issue has been the main drive of our research. We aim to produce a model for hate speech detection posted on a social media platform. The data was obtained from github– a hosting provider, consisting of tweets. Word2vec was employed as the method for feature extraction while support vector machine (SVM) with RBF as kernel function was used for data classification. The model was built and tested with a 70:30 ratio of data training and testing, in which we achieved the highest accuracy level of 85% with the settings of gamma= 0.1 and C-value= 10. The accuracy dropped to 69.7% when the model was tested with different datasets. With the development of hate speech detection models, we are optimistic towards a better society where social media users are less prone to negatively-intended information spread.**

Keywords : hate speech, support vector machine, word2vec, social media, tweets.

I. INTRODUCTION

Social media is no longer seen as such a tertiary or privilege as a decade before. It is used in every layer of society and it shares everything from a user posting on it to numerous others. Hootsuite, a platform of media social management and a "We are Social" marketing agency, in Januari 2021 exposed that about 61.8% of 274.9 million Indonesian use social media [1]. The minister of Indonesia's Ministry of Communication and Informatics, Johnny G. Plate, mentioned Microsoft's digital survey to measure the level of civility in 2020 revealed that Indonesia was the 29th out of 32 countries and was also the lowest position among SEA countries. The survey was measured from the spread of hoax, hate speech, discrimination, cyberbullying, and such [2]. It was revealed that there were more than 3,640 cases of racial hate speech on social media from 2018 to April 2021 [3]. Hate speech itself is a deteriorating behavior that causes a direct or indirect attack to another person or group [4][5][6][7]. Hate speech is words or texts expressed by an individual or a group to provoke anger or hate from a targeted individual or group. It is aimed to attack the target directly or indirectly, expecting to incite a negative or violent feedback from the target [8].

Despite the existence of virtual police monitoring the content posted on social media and punishing the law violator(s) [9], the fact is that the count of cases did not seem to decelerate. From February 23rd to April 12th 2021, there were 329 contents flagged by the virtual police with Twitter topped at the chart of most prevalent platforms (195 contents) [10].

Machine learning method has been widely applied in dealing with hate speech on social media. Universally, machine learning is applied to detect whether a sentence contains the element of hate speech or not [4-7, 11-12]. Previous research stated that Twitter is the most-used platform to spread hate speech whereas the most popular topic revolved around politics and election, analyzed by employing SVM as classification method [11].

Several researchers had employed SVM in the detection of hate speech on Twitter [4-7, 11-14]. The research investigating Indonesian tweets using SVM had 74.88% accuracy while still struggling with misclassification issues regarding swear words containing animals' names [4]. Another analysis on Indonesian tweets by using SVM method combined with Term Frequency Inverse Document Frequency (TF-IDF) as vectorizers performed 87.07% in accuracy [12], which is higher than other machine learning techniques. Another research on Twitter text in English evidently showed that the integration of SVM with TF-IDF would result in better performance than other machine learning techniques, in which it achieved up to 96.68% accuracy when applying pre-processing techniques. Applying pre-processing techniques significantly enhanced the accuracy. A notable conclusion was that pre-processing techniques overcame class imbalance issues in the tested dataset [5]. The result of a hate-speech detection research conducted in South Africa also proves that SVM method can predict word classification better than other method, in which the optimized SVM method with character n-gram yielded 64.6% accuracy [13].

Another hate speech research was conducted to analyze Turkish tweets. The work added another positive comment on the performance of SVM method amongst other machine learning techniques, despite the common drawback of unique symbols or some English words or slangs detection [7].

Information extraction is one critical stage in determining whether a sentence is categorized as hate speech or not. In this stage, a sentence will be deconstructed into a group of words that construct the sentence itself and the meaning of each word is interpreted. Word2vec is a very popular method in text analysis for its ability to interpret words in varying language and dataset with relatively higher accuracy [14][15].

There were studies on hate speech detection by employing SVM method integrated with word2vec as the

feature extraction tool. Word2vec model firstly introduced by Mikolov et al. Word2vec represents the learned associated words to observe semantic relationship, enhancing NLP *(Natural Language Processing)*. There are two types of word2vec, Skip-gram and CBOW [16]. The research results suggested that word2vec could improve the accuracy of the model when integrated with SVM method [14][15]. However, exercising word2vec requires data training steps with a diverse range of words to realize a high accuracy level [17].

Based on those conditions, this work designed a classification model with the implementation of word2vec for hate speech in Indonesian, following SVM method. We expect that the application of word2vec as the extraction model while exercising SVM with RBF kernel function may result in better accuracy in predicting textual hate speech in Indonesian.

II. METHODS

Our work exercised feature extraction by using word2vec with SVM method to classify Indonesian words. Fig. 1 displays the workflow of this study whereas Fig.2 exhibits the process of building the model based on the feature extraction. As Fig. 1 shows, this study starting with literature review process to find out previous research on hate speech detection. Continued with the data collection process for model development and testing. After the datasets is obtained, it will be applied pre-processing to the datasets prior to developing word2vec model. After the feature extraction, the result would be the input of classification stage for data training and data testing. Data training was performed to develop reliable SVM model for hate speech prediction. The trained SVM model would be evaluated in the data testing stage. The performance of the SVM model was measured by using the confusion matrix, identifying its accuracy, precision, recall, and F1-score.

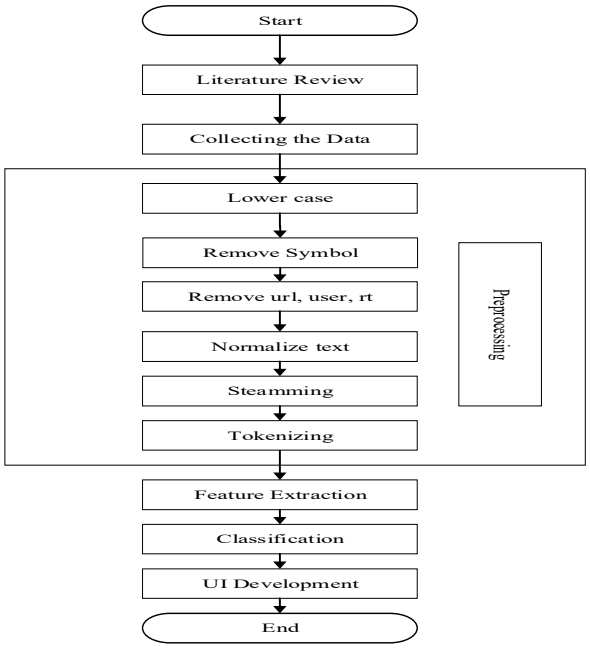

Fig. 1. Research Framework

In Figure 2 can be seen the integration process between the results of preprocessing with feature extraction using the word2vec model. The result of feature extraction are then used as input for classification process with SVM. From word2vec model, the mean of embedded word vector will be generated as can be seen in Table III. The classification process will produce prediction results whether a sentence contains hate speech or not. The results of these predictions will be compared with ground truth to measure the performance of the resulting model.

A. Data Source

The data used in this research sourced from 2 websites. The first one was taken from https://github.com/okkyibrohim/id-multi-label-hate-speech-and-abusive-language-detection. This dataset consisted of 13,169 instances of tweets, classified into 2 classes: hate-speech class (7,608 instances) and non-hate-speech class (5,561 instances). The second dataset was obtained from https://github.com/ialfina/id-hatespeech-detection. It contained only 713 instances, consisting of 453 instances of hate-speech class and 260 instances of non-hate-speech class. The first dataset was used for the training and testing process. Meanwhile, the second dataset was used solely for data testing purpose, to measure the performance of the model when analyzing different datasets from the one used in the model's development. Both datasets were acquired from tweets, considering its infamy as the most-used platform to spread hate speech [10-11].

B. Preprocessing

In this phase, the data as texts were prepared into a specific format to simplify the classifier to detect and categorize words into hate or non-hate speech classes. The

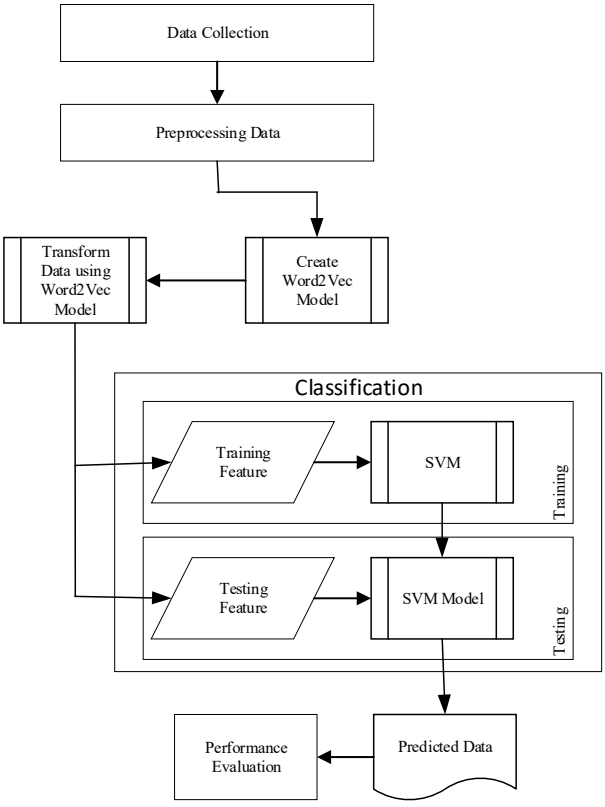

Fig. 2. Integration of Word2vec and SVM

preprocessing included:

- Lower case, is a process to converting all letters into lower case letters.
- Remove symbol, that means removing 27 symbols from the data: !@#$%?^&*-:.,/\\'|;\]\[\"()
- Remove url, user, rt or in the other words will removing texts containing url, the username and the text "rt".
- Normalize text that will converting miss-typed or misspelled words into their formal and correct form according to the available library of dictionary [18], e.g., the word *aamiin* (aameen) would be normalized into *amin* (amen).
- Steaming process that converting words into its root/base form and removing their affix(es), e.g., the word "freed" would be converted into "free".
- Tokenizing, is a process to deconstructing a sentence into a group of its composing words, e.g., the sentence "aameen hopefully Indonesia would be freed from corona" would be destructed into six words, i.e. "amen", "hope", "indonesia", "free", "from", "corona".

C. Feature Extraction

The next step was feature extraction by using the word2vec method and the skip-gram model. In this stage, the meaning of the composing word was identified. We implemented the Gensim library with parameters such as window, size, min_count, sg, negative, and epoch. Parameter "sg" indicates the model used is skip gram. Parameter "window" is the range between the current and predicted word within a sentence while parameter "size" determines the dimension. Parameter "min_count" is the minimal number of words extracted from the text that will be trained. The value of each parameter was obtained from the trial and error process to achieve the highest accuracy possible, i.e. window = 5, min_count = 1, size = 300, sg = 1, negative = 5.

D. Classification

After building the word2vec model, we developed the model to classify data by employing SVM method with RBF kernel function. Parameters of C and Gamma are influential in SVM with the RBF kernel [19].

E. Design of User Interface

A UI was built to help users in the process of hate speech detection by using React Js with Python Flask for the API. This interface facilitates users to do an initial check on the sentences to be posted in a social media platform.

III. RESULT AND DISCUSSION

A. Building Detection Model

Prior to classification, several steps of word2vec method were applied:

- Create the Vocabulary out of the sentences in the datasets and randomly generate numbers to fill the matrix W1 based on the determined dimension. This research used a dimension of 300. Table I is an example of 5-dimensional matrix W1 for the sentence "I love Indonesia".
- Run the training process by using the skip-gram model, consisting of seven steps:
  i. Generate matrix W2, a transpose of matrix W1.
  ii. Determine the word to be the input and target, e.g., in the sentence of "I love Indonesia" with window = 1 and the input = love, the target will be 1 word surrounding the input, which in this case are "I" and "Indonesia". Then, transform the word into numerical data by one-hot encoding.
  iii. Calculate the hidden layer by multiplying the input of one-hot encode of the input word with matrix W1. One-hot encode input of the input word (love) is [0, 1, 0, 0], multiplying it by matrix W1 results in hidden layer matrix [0,926865 0,850752 0,041328 0,266075 0,381857].
  iv. Calculate the output by multiplying the hidden layer with matrix W2. The result is [1,423588925 1,801176371 0,973013499 1,634579612].
  v. Calculate the softmax and error of the output prediction for the words "I" and "Indonesia". The result is [0,230897652 0,33682449 0,147142149 0,285135708]. Subtract the one-hot-encoded value of the target word, which is [1, 0, 0, 0] for the word "I" and [0, 0, 1, 0] for the word "Indonesia" from the result. Then sum all of the error values.
  vi. Update matrix W2 by multiplying the hidden layer with total error and further multiply the result with learning rate. Subtract the old value of matrix W2 from the new one to result in the update value.
  vii. Update matrix W1 by multiplying matrix W2 with total error and the one-hot encoded value of input word– which is "love", and multiply it with learning rate value. The updated W1 value is obtained by subtracting the old value from the new one.

After obtaining the word vector for each word, the sentence was transformed into vectors by using the word2vec method. The sentence "I love Indonesia" was vectorized by the process of embedding and mean calculation, with the results shown in the following Table II and Table III.

TABLE I. W1 MATRIX

Word	D1	D2	D3	D4	D5
I	0,244662	0,78653	0,045442	0,92994	0,728981
love	0,926865	0,850752	0,041328	0,266075	0,381857
Indonesia	0,084806	0,443918	0,864548	0,67002	0,79281

TABLE II. EMBEDDING WORD VECTOR

Words	Word2vec vector
I	[0.12 0.13, 0.15 0.15 0.62]
love	[0.1 0.5, 0.5 0.7 0.25]
Indonesia	[0.2 0.7, 0.3 0.4 0.44]

TABLE III. MEAN OF EMBEDDED WORD VECTOR

Words	Word2vec vector
I	[0.12 0.13, 0.15 0.15 0.62]
love	[0.1 0.5, 0.5 0.7 0.25]
Indonesia	[0.2 0.7, 0.3 0.4 0.44]
Total	[0.32 0.25 0.23 0.26 1.31]
Mean	**[0.106 0.083 0.076 0.866 0.436]**

The mean value would be the vector of the sentence *"*I love Indonesia" and would be used as the feature of the sentence in the classification process.

We performed trial and error on various combinations of value sets to determine the parameters for the best result of a classification model using SVM, implementing the Sklearn Library package. The process can be examined in Table IV, in which the best accuracy is achieved at Gamma= 0.1 and C= 10. These values were set as the benchmark throughout the whole experiment. To get the best value for Gamma and C, we conduct several experiments using variants of C and Gamma values, and took the C and Gamma values that give the best accuracy value.

B. Model evaluation

Model testing was conducted by splitting the first dataset into two groups, which were for training and testing with a ratio of 7:3 [6].

Table IV displays the highest accuracy (85%) achieved by setting the values of C = 10 and gamma = 0.1. These values were used for building the SVM model. The testing on accuracy was completed by employing the confusion matrix, measuring the values of True Positive, True Negative, False Positive, and False Negative between the outputs of the model and the ground truth. Confusion matrix was applied for both datasets used in this work. The confusion matrix of the first dataset can be seen in Table V. From Table V, we can see that there are 1990 True Positive, which means that there are 1990 same predictions between the model and the ground truth that state the sentences as hate speech. 266 False Positive means that there are 266 unequal prediction results between the model and the ground truth. Where the model predicts hate speech but ground truth states otherwise. 1294 True Negative means that both the model and the ground truth state the sentences are not-hate speech. 306 False Negative is the opposite of False Positive, where the model state the sentence as not-hate speech, but the ground truth state the opposite.

This work ran an experiment on two different datasets for testing. The first scenario was building the model and testing it with the same dataset, which was the first dataset. The second scenario was that the model built in the first scenario performed a test on the second dataset. The results of this experiment informed that there was a decrease in accuracy when the model was to evaluate different dataset.

TABLE IV. HYPER PARAMETER TESTING

C	Gamma	Accuracy	Precision	Recall	F1
10	**0.1**	**0,85**	**0,83**	**0,81**	**0,82**
10	0.01	0,82	0,8	0,77	0,78
10	0.001	0,8	0,79	0,72	0,75
50	0.1	0,83	0,8	0,79	0,79
50	0.01	0,83	0,81	0,78	0,79
50	0.001	0,81	0,79	0,75	0,77
100	0.1	0,82	0,79	0,78	0,79
100	0.01	0,84	0,81	0,79	0,8
100	0.001	0,82	0,8	0,76	0,77

TABLE V. CONFUSION MATRIX OF THE FIRST DATASET

		Actual	
		Yes	No
Prediction	Yes	1990	266
	No	306	1294

Table VI describes the details of the results. The coloum Dataset 1 in Table VI express the value of accuracy, precision, recall, and F1-score that gained when the model was training and testing using the same dataset, namely first dataset. Meanwhile the coloumn Dataset 2 express the value when the model was training using first dataset and testing using second dataset.

The performance dropped 15% in accuracy due to the fact that several words in the second dataset for data testing did not acquire their vector. Word2vec with skip-gram model requires a large number of word variations. Meanwhile, the SVM method needs a varying dataset to improve its performance in accuracy.

The next testing scenario involved random sentences in Bahasa Indonesia in the evaluation. The purpose of this test was to evaluate the model's ability to analyze varying word combinations, basic word placement, and words with (an) affix(es). Table VII shows the results of this test.

As seen in Table VII, the model predicts a sentence as a hate speech if the word "dog" is accompanied by the word "such a" or "what a" and/or a name (andi) or a subject "*loe"*. When there was a negation (the word "not") changing its meaning, the system labeled it as "Non hate speech". However, there was an error when labeling the sentence "Andi's dog" owing to the steaming process which deformed the word *"*the dog of" into its basic word *"*dog", completely altering the meaning of the original word.

C. User Interface

Fig. 3 displays the UI developed by the implementation of React js with Python flask. This page is accessible at http://ui-ta.herokuapp.com/.

TABLE VI. EVALUATION OF BOTH DATASETS

Parameter	Dataset 1	Dataset 2
Accuracy	85%	69.7%
Precision	83%	64%
Recall	81%	91%
F1-score	82%	75%

TABLE VII. THE RESULT OF TESTING ON RANDOM SENTENCE

Sentence	Detection result
You're such a dog	Hate Speech
Dog is an animal	Non Hate Speech
Dog	Non Hate Speech
What a dog	Hate Speech
You dog	Hate Speech
You are not dog	Non Hate Speech
Andi's dog	Hate Speech

Fig. 3. Webpage preview of hate-speech detection

IV. CONCLUSION

Based on the results, it is concluded it is reliable to apply word2vec method for feature extraction and the integrated SVM method with RBF kernel function as the hate-speech classifier in developing hate-speech detection model. We produce a model with 85% from analyzing 13,169 data with a ratio of data training:data testing = 7:3. This result is achieved with the value settings as follows: dimension= 300, C= 10, and gamma= 0.1. However, the accuracy lowers by 15% when the data training and data testing are inputted with different datasets. The experimented discover a disadvantage in the model, in which it erroneously detected words with (an) affix(es) and positioning of basic word. For further advancement, it is recommended to provide larger numbers of data training and employ other word embedding techniques like Glove and Fasttext.

REFERENCES

[1] [1] C. Stephanie, "Riset Ungkap Lebih dari Separuh Penduduk Indonesia _Melek_ Media Sosial," *tekno.kompas.com*, 2021. [Online]. Available: https://tekno.kompas.com/read/2021/02/24/08050027/riset-ungkap-lebih-dari-separuh-penduduk-indonesia-melek-media-sosial.

[2] [2] Agregasi VOA, "Netizen Indonesia Dinilai Kurang Beradab, Pemerintah B entuk Komite Etik Internet," 2021. [Online]. Available: https://techno.okezone.com/read/2021/02/28/16/2369605/netizen-indonesia-dinilai-kurang-beradab-pemerintah-bentuk-komite-etik-internet.

[3] [3] "Sejak 2018, Kominfo Tangani 3.640 Ujaran Kebencian Berbasis SARA di Ruang Digital," 2021. [Online]. Available: https://www.kominfo.go.id/content/detail/34136/siaran-pers-no-143hmkominfo042021-tentang-sejak-2018-kominfo-tangani-3640-ujaran-kebencian-berbasis-sara-di-ruang-digital/0/siaran_pers.

[4] [4] K. M. Hana, Adiwijaya, S. Al Faraby, and A. Bramantoro, "Multi-label Classification of Indonesian Hate Speech on Twitter Using Support Vector Machines," *2020 Int. Conf. Data Sci. Its Appl. ICoDSA 2020*, pp. 6–12, 2020.

[5] [5] B. Pariyani, K. Shah, M. Shah, T. Vyas, and S. Degadwala, "Hate speech detection in twitter using natural language processing," *Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mob. Networks, ICICV 2021*, no. January 2018, pp. 1146–1152, 2021.

[6] [6] G. Koushik, K. Rajeswari, and S. K. Muthusamy, "Automated hate speech detection on Twitter," *Proc. - 2019 5th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2019*, pp. 2019–2022, 2019.

[7] [7] R. B. Sahi, Havvanur; Kilic, Yasemin; Sagham, "Automated Detection of Hate Speech towards Woman on Twitter," in *3rd International Conference on Computer Science and Engineering*, 2018, pp. 533–536.

[8] [8] L. S. Widayati, "UJARAN KEBENCIAN: BATASAN PENGERTIAN DAN LARANGANNYA," *INFO Singk.*, vol. X, no. 06/II/Puslit/Maret/2018, 2018.

[9] [9] "Warga +62 di Bayang-bayang Pengawasan Polisi Virtual," *CNN Indonesia*, 2021. [Online]. Available: https://www.cnnindonesia.com/nasional/20210317112117-12-618528/warga--62-di-bayang-bayang-pengawasan-polisi-virtual.

[10] [10] F. M. Sidik, "329 Konten Medsos Ditegur Virtual Police, Terbanyak di Twitter," 2021. [Online]. Available: https://news.detik.com/berita/d-5535129/329-konten-medsos-ditegur-virtual-police-terbanyak-di-twitter.

[11] [11] Rini, E. Utami, and A. D. Hartanto, "Systematic Literature Review of Hate Speech Detection with Text Mining," *2020 2nd Int. Conf. Cybern. Intell. Syst. ICORIS 2020*, pp. 1–6, 2020.

[12] [12] U. A. N. Rohmawati, S. W. Sihwi, and D. E. Cahyani, "SEMAR: An interface for Indonesian hate speech detection using machine learning," *2018 Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2018*, pp. 646–651, 2018.

[13] [13] O. Oriola and E. Kotze, "Evaluating Machine Learning Techniques for Detecting Offensive and Hate Speech in South African Tweets," *IEEE Access*, vol. 8, pp. 21496–21509, 2020.

[14] [14] R. N. Waykole and A. D. Thakare, "a Review of Feature Extraction Methods for Text Classification," *Int. J. Adv. Eng. Res. Dev.*, vol. 5, no. 04, pp. 351–354, 2018.

[15] [15] S. Al-Saqqa and A. Awajan, "The Use of Word2vec Model in Sentiment Analysis: A Survey," *ACM Int. Conf. Proceeding Ser.*, pp. 39–43, 2019.

[16] [16] X. Rong, "word2vec Parameter Learning Explained," pp. 1–21, 2014.

[17] [17] M. A. Fauzi, "Word2Vec model for sentiment analysis of product reviews in Indonesian language," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 1, p. 525, 2019.

[18] [18] M. O. Ibrohim and I. Budi, "Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter," pp. 46–57, 2019.

[19] [19] A. C. Müller and S. Guido, *Introduction to with Python Learning Machine*. 2017.