Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study

Conference Paper · October 2017

DOI: 10.1109/ICACSIS.2017.8355039

4 authors, including:

Ika Alfina
University of Indonesia
19 PUBLICATIONS **117** CITATIONS

SEE PROFILE

Rio Mulia
University of Indonesia
1 PUBLICATION **26** CITATIONS

SEE PROFILE

Mohamad Ivan Fanany
University of Indonesia
140 PUBLICATIONS **710** CITATIONS

SEE PROFILE

Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study

Ika Alfina, Rio Mulia, Mohamad Ivan Fanany, and Yudo Ekanata
Machine Learning and Computer Vision Laboratory
Faculty of Computer Science Universitas Indonesia
Depok, Indonesia
ika.alfina@cs.ui.ac.id, rio.mulia@ui.ac.id, ivan@cs.ui.ac.id, yudo.ekanata51@ui.ac.id

***Abstract*—The objective of our work is to detect hate speech in the Indonesian language. As far as we know, the research on this subject is still very rare. The only research we found has created a dataset for hate speech against religion, but the quality of this dataset is inadequate. Our research aimed to create a new dataset that covers hate speech in general, including hatred for religion, race, ethnicity, and gender. In addition, we also conducted a preliminary study using machine learning approach. Machine learning so far is the most frequently used approach in classifying text. We compared the performance of several features and machine learning algorithms for hate speech detection. Features that extracted were word n-gram with n=1 and n=2, character n-gram with n=3 and n=4, and negative sentiment. The classification was performed using Naïve Bayes, Support Vector Machine, Bayesian Logistic Regression, and Random Forest Decision Tree. An F-measure of 93.5% was achieved when using word n-gram feature with Random Forest Decision Tree algorithm. Results also show that word n-gram feature outperformed character n-gram.**

Keywords—building dataset; classification; hate speech detection; machine learning

I. INTRODUCTION

Nowadays, the number of social media users is increasing rapidly. Facebook as the market leader, on June 2017 had 2 billion monthly active users[1], which is more than a quarter of human population on earth. This shows that social media has become an important communication medium today. Social media technology enables the message to be sent quickly, become widespread and even viral if the topic attracts public attention. Unfortunately, this also means that hate speech can also spread easily and quickly that it can lead to conflicts between groups in society.

Hate speech is "*any communication that disparages a person or a group on the basis of some characteristic such as race, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic*" [1]. In [2], the work proposed 11 criteria for detecting hate speech, some of which are: uses of a sexist or racial slur, attack a minority, promotes hate speech or violent crime, blatantly misrepresents truth, shows support of problematic hashtags, defends xenophobia or sexism, and contains a screen name that is offensive.

Hate speech, especially concerning race and religion, became the most reported form of online crime in 2016, according to Indonesian police[2]. The police officer in Indonesia claimed at least 5 cases reported each day, which means there are about 150 each month[3]. The police also said that handling cyber criminal is not easy that facilities and human resources are needed. This makes automatic hate speech detection is necessary to be developed for the Indonesian language so that the police can detect the spread of hate speech quickly.

There are several studies that have proposed automatic hate speech detection in English [2]–[4]. These three works used machine learning approach and used Twitter as the source of the dataset. The special of Twitter is that it provides API to retrieve its tweets so that we can use those tweets as research data.

Twitter is one of the popular social media these days with more than 300 million monthly active users[4]. Twitter prohibits users to post violent threats, harassment, and hateful contents[5]. However, there are still tons of users who disobey the rules and use their Twitter account to spread hate speech and negative words.

In [3], they worked on hate speech detection against black people using word unigram feature and Naïve Bayes (NB) algorithm. In [4], the study compared the performance of three machine learning algorithms: 1) Bayesian Logistic Regression (BLR); 2) Random Forest Decision Tree (RFDT); and 3) Support Vector Machine (SVM) for cyber hate detection against race, ethnicity, and religion using word unigram and bigram features. Reference [2] implemented the character n-gram and word n-gram as the features and used BLR as the classification algorithm to detect hate speech in general.

As far as we know, the study of hate speech detection in the Indonesian language is still very rare that we found only one previous work [5] on this subject. In [5], the study focused on hate speech against religion. This work built a new dataset with two labels: hate-speech-against-religion and not, built the hate-speech dictionary, and compared the performances of several

[1] https://en.wikipedia.org/wiki/Facebook

[2] http://www.thejakartapost.com/news/2017/03/26/hate-speech-clouds-indonesias-internet-in-2016-police.html

[3] http://www.tribunnews.com/metropolitan/2017/05/31/setiap-hari-polda-metro-jaya-terima-lima-laporan-kasus-ujaran-kebencian

[4] https://en.wikipedia.org/wiki/Twitter

[5] https://support.twitter.com/articles/18311

features when combined with NB and SVM. Although the F-measure obtained on this work reached 90%, it was found that the quality of the dataset was inadequate. The tweets with the label non-hate-speech-against-religion were generally unrelated to religion. The unbalanced number of tweets related to religion and not in this class has made the resulting classifier wrongly detected tweets related to religion as hate speech.

In our work, we created a new dataset for hate speech detection in the Indonesian language that covers hate speech in general such as hatred of religion, ethnicity, race, and gender. We also conducted a preliminary study to find out which combination of machine learning algorithm and feature gave the best result. The contributions of our work are:

- Creating a new dataset for hate speech detection study in the Indonesian language. This dataset has been made public for the next works[6].
- Presenting a preliminary performance benchmark when using machine learning approach. We showed which combination of feature and classification algorithm that worked best for our dataset.

This paper consists of 5 sections and is organized as follows. Section 2 describes related works in hate speech detection studies. Our methodology is explained in Section 3. We discuss experiments and results in Section 4 and finally, Section 5 contains conclusions and future work.

II. Related Works

Several works have studied hate speech detection in English [2]–[4]. These studies used machine learning approach and Twitter as the data source. In [3], the study focus on hate speech detection against black people. They used Naïve Bayes algorithm and word unigram as the feature. In labeling the dataset, they assigned three annotators that consist of people who have a different racial background to improve the objectivity.

Reference [4] did not focus on one issue but more general that covers ethnic group, race, ethnicity, religion, etc. Bag of words technique was also implemented, using unigram and bigram features. This study shows that BLR, RFDT, and SVM have the same performance to detect hate speech on tweets in English, which is 77% on the F-measure score.

Meanwhile, [2] aimed to compare features that are suitable to detect hate speech in English. They chose word n-gram and character n-gram as the main features. Word n-gram feature that used was the combination of unigram and bigram. In character n-gram, each sentence is considered as a bag of character n-grams, in which every attribute in features is a string with the length of *n* [6]. For example, the character 4-grams of "hate speech" will be extracted as |hate|, |ate_|, |te_s|, |e_sp|, |_spe|, |spee|, |peec|, and |eech|. Moreover, [2] also used gender and location as additional features. BLR was chosen as the classification algorithm. The results showed that character n-gram outperformed the word n-gram feature for detecting hate speech in English with 10% difference in accuracy. Besides of that, gender and location did not have any effect towards the algorithm classification performance.

In [5] that worked on hate speech detection against religion in the Indonesian language, the features used were word unigram and bigram, the number of hateful words and hateful phrases, and the number of words having negative sentiment. The algorithms to be compared were NB and SVM. In order to count the number of words/phrases related to hate speech, this work also built the hate-speech dictionary. Due to the unbalanced number of religion-related tweets and not in the non-hate-speech-against-religion class, the resulting dictionary had a poor quality since it was more appropriate as the dictionary of religion-related words/phrases than as the hate speech dictionary.

In our research, we adopted several approaches used by [2]–[5]. We chose the general topic as [2] and [4] did. From [3], we adopted the way they annotated the dataset that involved annotators from the different background to improve the objectivity. As features, we used unigram as used by [2]–[5], bigram as used by [2], [4] and [5], character trigram (character 3-grams) and character quadragram (character 4-grams) as used by [2], and finally the negative sentiment as used by [5]. We compared the performance of four classification algorithms, i.e. NB, BLR, RFDT, and SVM.

III. Methodology

In this section, we discuss how to create the dataset and the methodology in conducting hate speech detection using machine learning approach.

A. Creating the Dataset

The process of creating the dataset consists of two main steps, collecting and annotating the dataset.

1) Data Collection

We used Twitter data as the source of the dataset and collecting the tweets using Twitter Streaming API[7]. The tweets were related to a political event, the Jakarta Governor Election 2017. This election was a potential source of hate speech data because one of its candidates came from a minority group in Indonesia, in terms of religion and race, while another candidate was a woman that potentially triggered hate speech related to gender.

The election took two rounds. We gathered the tweets from the beginning of February 2017 which was the first election round until the second election round which took place on April 2017. Some keywords that related to that election were used, e.g. "#DebatPilkadaDKI", "#SidangAhok", "Pilkada Jakarta 2017", etc. We managed to collect around 40,000 tweets. After removing the duplicated tweets, we had 1,100 tweets to be labeled manually.

2) Data Annotation

Each tweet on the dataset will be labeled whether it contains hate speech or not. There are only two labels. Tweets

[6] https://github.com/ialfina/id-hatespeech-detection

[7] http://apps.twitter.com

containing hate speech will be labeled as "HS", and those which are not labeled as "Non_HS".

The dataset was annotated manually by 30 volunteers who were all college students in Jakarta and surrounding areas with the age range of 17 - 24 years. Consists of 43.3% of men and 56.7% of women. Fig. 1 shows the distribution of volunteers by religion and Fig. 2 displays distribution by race/ethnicity. It can be seen that the background of volunteers was very diverse in terms of gender, religion and race/ethnicity. This needs to be done to reduce the bias because the nature of the data to be annotated is quite subjective.

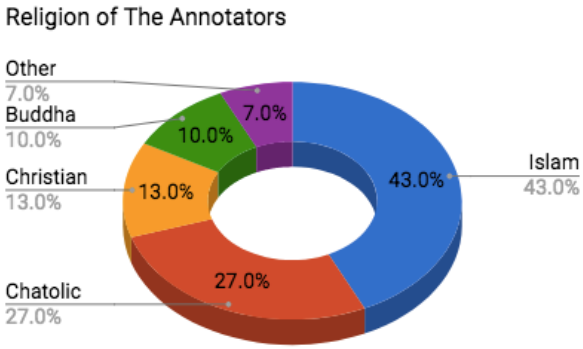

Fig. 1 The distribution of the religion of the annotators

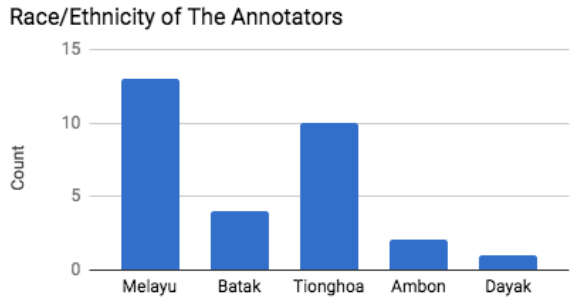

Fig. 2 The distribution of the race/ethnicity of the annotators

We divided 1,100 previously collected tweets into 22 sets, each consisted of 50 tweets. Each set would be annotated by three people from different religious, racial and gender backgrounds. Each volunteer may annotate more than one set of data. Before volunteers began to annotate, we ensured that they understand the definition of hate speech we use in this study.

For each tweet, respondents should answer whether the tweet contains hate speech or not. The final label of a tweet is determined by the number of "yes" it gets. If the number is 3 then the tweet will be labeled as "HS", while if the number is 0 it will be labeled as "Non_HS". The tweets with the number of yes of 1 or 2 were removed from the dataset because it was considered ambiguous. In another word, the annotator agreement of each tweet should be 100%.

Of 1,100 tweets, only 713 tweets had 100% agreement, consist of 260 tweets as "HS" and 453 tweets as "Non_HS". Thus, the annotator agreement for 1,100 tweets was 64.8%. Since the number of tweets with label "HS" did not equal with the number of tweets with "Non_HS" label, the resulting dataset until this stage was an unbalanced dataset.

In [7], it was explained that unbalance dataset can cause a negative effect on classification performance since the imbalance number of dataset between majority and minority class tends to make the majority class has a better performance than the minority one. So, we decided to transform our original dataset into a balanced dataset using an under-sampling method. We kept all our 260 HS tweets into the new dataset and choose randomly 260 of 445 tweets of Non_HS tweets. Our final dataset became a balanced dataset with the size of 520.

B. Hate Speech Detection

Another objective of our research is to compare features and machine learning algorithms to find out which combination of features and algorithm that have the best performance. Our methods consist of three stages: 1) preprocessing; 2) feature extraction; and 3) classification and evaluation.

1) Preprocessing

We adopted the preprocessing method used by [8] with little modification. There are six steps in the preprocessing stage, i.e. 1) retweet removal; 2) text cleansing; 3) lowercasing; 4) spell correction; 5) negation handling; and 6) stopword removal. The only step in [8] that we did not carry out was hashtag handling. We decided to threat the hashtag as the ordinary word.

2) Features Extraction

We used the bag of words (BOW) model [9] in representing the text. In general, we utilized 3 classes of features: word n-gram, character n-gram, and negative sentiment. For word n-gram, we implemented only for n=1 (word unigram) and n=2 (word bigram). For character n-gram, we implemented only for n=3 (character trigram) and n=4 (character quadragram). The usage of character n-gram was based on [2]. For the negative sentiment feature, we adopted the method used by [5] that used sentiment dictionary created by [10] as the basis in counting the number of words in a tweet that has negative sentiment. Thus, we used five features: word unigram, word bigram, character trigram, character quadragram, and negative sentiment.

3) Classification and Evaluation

We used supervised learning approach in detecting hate speech in the Indonesian language. We would compare the performance of four algorithms: NB, SVM, BLR, and RFDT using our dataset.

We used Weka[8] to conduct the experiments. The evaluation was carried out using the 10-fold cross validation method. In this paper, only the weighted average F-measure for all class is reported due to the limited space.

[8] http://www.cs.waikato.ac.nz/ml/weka/

IV. Experiments and Results

In this section, we discuss the experiment results and analysis.

A. Experiment Results

We conducted three scenarios of experiments. First, we compared the performance of every combination of features and algorithms. Second, we compared the performance of word n-gram vs. character n-gram. Third, we compared the performance of each algorithm when using all the five features altogether.

Table I shows the classification result for each combination of four machine learning algorithms and five features in terms of the weighted average F-measure. The best F-measure of 82.6% was achieved when using the combination of character quadragram and RFDT, closely followed by the combination of unigram and BLR or RFDT, and the combination of character trigram and RFDT. These four best results are in a narrow range of [81.7%, 82.6%] so that it can be said that the four combinations are almost equally superior.

TABLE I. The F-measures for Each Feature and Algorithm

Feature	Weighted Average F-measure (%)			
	NB	SVM	BLR	RFDT
Word unigram	79.8	74.6	**81.7**	**81.7**
Word bigram	66.5	64.5	74.5	66.7
Char trigram	77.6	72.8	70.8	**81.7**
Char quadragram	76.8	68.4	80.4	**82.6**
Negative Sentiment	55.7	67.1	67.1	67.1

Among the five features, word unigram was consistent to be the most predictive feature, while word bigram or negative sentiment alternately became the least predictive one. Character quadragram performed better than character trigram when combined with BLR and RFDT, but the reverse when combined with NB and SVM. In general, F-measure of character n-gram was higher than word bigram and negative sentiment, but lower than word unigram.

Algorithm NB and SVM had their highest result when combined with unigram. BLR worked best when using word unigram or character quadragram, and RFDT became the best feature when combined with word unigram or character n-gram. SVM had the lowest performance among the four algorithms, except when combined with character trigram and negative sentiment.

Table II shows the comparison of the weighted average F-measure between word n-gram and character n-gram. In our work, word n-gram feature was the union of word unigram and bigram, while character n-gram was the union of character trigram and quadragram. We can see that word n-gram always outperformed character n-gram. This result was contradictory with the result of [2] that reported character n-gram was better than word n-gram. The best result of F-measure of 93.5% was achieved when word n-gram feature was combined with RFDT. Among the fours, SVM had the lowest results.

TABLE II. The F-measures of Word n-gram vs. Character n-gram

Feature	Weighted Average F-measure (%)			
	NB	SVM	BLR	RFDT
Word N-Gram	90.2	86.5	91.5	**93.5**
Char N-Gram	79.4	73.0	78.1	84.2

Table III shows the comparison of the weighted average F-measures between the four algorithms when using all five features altogether. RFDT was superior with F-measure of 89.8% and SVM again had the lowest performance of 72.3% of F-measure. However, the highest score in this third scenario was less than the highest score in the second scenario but better than the highest scores in the first scenario.

TABLE III. The F-measures Using All Features

Feature	Weighted Average F-measure (%)			
	NB	SVM	BLR	RFDT
All	82.5	72.3	86.0	**89.8**

Word unigram feature in the first scenario had the best performance of 81.7%. Adding word bigram to the features set had improved the F-measure in a margin around 12%. However, if we added the other three features altogether then the F-measure just increased about 8% compared to the first scenario. We suggested that using word n-gram only was preferable while reducing the cost of extracting features for character n-gram and negative sentiment.

B. Discussion and Limitation

Based on the experiment results, the word n-gram outperformed the character n-gram feature. The result was not the same as [2] who said the character n-gram was better when detecting hate speech in English. Among word n-gram, word unigram was always superior to word bigram and among character n-gram, character quadragram frequently outperformed character trigram.

The negative sentiment feature often had the lowest performance among the five features. Although in our work we only use the number of words with negative sentiment and in [8] they used both the positive and the negative ones, the result was similar. We suggest that this sentiment feature was either less predictive or the sentiment dictionary we used that originally built for product review domain was not suitable for the dataset of the political domain.

The best performance of 93.5% of F-measure was achieved when word n-gram feature was combined with RFDT algorithm. This result was followed by the performance of word n-gram combined with BLR (91.5%) and NB (90.2%). If we compare these results with the combination of word unigram and BLR that gave the best result in the first scenario, we found that union of word unigram and word bigram was superior to word unigram alone.

In general, among the four algorithms, we can see that RFDT and BLR superior to both NB and SVM. These results are different with [4] that reported that RFDT, BLR, and SVM had the same performances in detecting hate speech in English.

Table IV shows two examples for each type error of the classification result: false positive and false negative. From the two examples of the false positive case, we suspected that the occurrence of word "ahok" (the name of the politician), "babi" (pig), or "penista agama" (religion humiliator) had made the classifier decide those tweets contained hate speech. Whereas in the false negative case, the absence of those terms made the classifier detected those sentences as non-hate-speech.

TABLE IV. THE EXAMPLES OF CLASSIFICATION ERROR

Result	Example
False positive	*Apakah perlu semua berita Ahok harus menggunakan 'penista agama'? Misalnya penista agama sedang kampanye di Jakarta pusat.* *(Is it necessary that all the news about Ahok must use the term 'religion humiliator'? For example, religious humiliator is campaigning in Central Jakarta)*
	Kemarin kan ada berita tentang penyakit meningitis dari babi, mksd gw apa nyambungnya sama Ahok? *(Yesterday there was news about the disease of meningitis from pig, I mean what is the relationship with Ahok?)*
False negative	*Huuu sylvi tak tahu apa-apa asal ngoceh. Keliatan bloonnya* *(Huuu sylvi does not know anything, talking without data. Showing that she is stupid)*
	Anies anda sadis. Topengmu terbuka lebar malam ini. Biar warga DKI yg menilai...apa yg anda katakan adalah pengecut *(Anies you are cruel. Your cover is exposed tonight. Let DKI citizens judge, what you say shows that you are a coward)*

After further observation to the dataset, we also found that the majority of tweets in hate-speech class was related to a candidate, "Ahok". Few data were associated with other candidates. This made the two examples of the false negative case that talking about other candidates ("Anis" and "Sylvi) were wrongly labeled by the classifier. The next study should pay more attention to the dataset proportion. Since the dataset was taken from an election event, hate speech against each candidate must be represented in the dataset. Both in hate-speech class and also in non-hate-speech class.

The classification error examples that shown in Table IV describe that the bag of words model that we used is inadequate to detect hate speech since this model only represents the occurrence of word/phrase. The sequence of occurrences and the context of words is not considered.

Although in this study we would like to cover hate speech in general, including hate speech against religion, ethnicity, race, and gender, in the dataset hatred toward gender and ethnicity were under-represented. Our data are generally about religious and racial hatred. This situation will also make the resulting classifier unable to detect hate speech related to gender and ethnicity.

V. CONCLUSION AND FUTURE WORK

In this research, we built a new dataset of tweets in the Indonesian language for hate speech detection and conducted a preliminary study by comparing the performance of several features and machine learning algorithms.

We manually annotated the tweets into two classes, tweets containing hate speech and not. The resulting dataset had a size of 520, consists of 260 tweets for each "hate-speech" and "non-hate-speech" class.

Based on the experimental results, the superior F-measure was achieved when using word n-gram, especially when combined with RFDT (93.5%), BLR (91.5%) and NB (90.2%). We found that word n-gram feature was superior to character n-gram. The results also showed that instead of using word unigram alone, it was better to union word unigram and word bigram. We also found that adding character n-gram and negative sentiment to the feature sets was not needed.

We also had different results with two previous works in hate speech detection in English. While [2] reported that character n-gram was better than word n-gram, we found the opposite. While [4] said that RFDT, BLR, and SVM had the same performance in detecting hate speech, we found out that SVM performance was much below RFDT and BLR.

For the future work on hate speech detection in the Indonesian language, we suggested two improvements. First, special attention to the dataset proportion should be conducted. Second, we found that the BOW model is inadequate to detect hate speech. We suggest using other methods that can figure out the semantics of the sentence.

REFERENCES

[1] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," *Proceeding LSM '12 Proc. Second Work. Lang. Soc. Media*, no. Lsm, pp. 19–26, 2012.

[2] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," *Proc. NAACL Student Res. Work.*, pp. 88–93, 2016.

[3] I. Kwok and Y. Wang, "Locate the Hate: Detecting Tweets against Blacks," *Twenty-Seventh AAAI Conf. Artif. Intell.*, pp. 1621–1622, 2013.

[4] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy and Internet*, vol. 7, no. 2, pp. 223–242, 2015.

[5] S. H. Pratiwi, "Detection of Hate Speech against Religion on Tweet in the Indonesian Language Using Naïve Bayes Algorithm and Support Vector Machine," *B.Sc. Tesis*, Universitas Indonesia, Indonesia, 2016.

[6] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos, "Words vs. character n-grams for anti-spam filtering," *Int. J. Artif. Intell. Tools*, vol. XX, no. X, pp. 1–20, 2006.

[7] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 4, pp. 42–47, 2012.

[8] I. Alfina, D. Sigmawaty, F. Nurhidayati, and A. N. Hidayanto, "Utilizing Hashtags for Sentiment Analysis of Tweets in The Political Domain," in *Proceedings of the 9th International Conference on Machine Learning and Computing*, 2017, pp. 43–47.

[9] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: A statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, no. 1–4, pp. 43–52, 2010.

[10] C. Vania, M. Ibrahim, and M. Adriani, "Sentiment Lexicon Generation for an Under-Resourced Language," *Int. J. Comput.*, vol. 5, no. 1, pp. 59–72, 2014.