Indonesia Hate Speech Detection using Deep Learning

Taufic Leonardo Sutejo and Dessi Puji Lestari
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
13514022@std.stei.itb.ac.id, dessipuji@informatika.org

***Abstract*—Hate speech brings negative impacts not only to the target victim, but also to the listener. The spread of hate speech can be done not only through the social media postings, but also through the video, campaign or speech. In this research, we develop models to detect hate speech in Indonesian Language from input text and speech by using deep learning approach. We utilized both textual and acoustic features and compare their accuracies. Experiments result showed that hate speech detection using only textual features is better than that of using acoustic features and both of combined features model. The best model using textual feature obtained F1-score 87.98% which is higher than the model of using acoustic feature only (F1-score 82.5%), and the model of using acoustic and lexical features (F1-score 86.98%).**

Keywords—hate speech; deep learning; Indonesian Language; textual features; accoustic features; multi-features

I. INTRODUCTION

The usage of internet in Indonesia has increased rapidly from year to year. *We are Social Company* researched that Indonesia is one of the highest internet consumptions in the world. One of those consumptions is the using of social media. Social media is an ideal place for internet users to share their daily life, give their thought about something, give away, and obtain information. Unfortunately, emerging problem arises due to the sharing and free access of information. The information spread in the social media can be positive information but also can be negative information.

Cyberbullying such as hate speech is one of the harmful or negative information spreading on social media. Spread of hate speech is carried by a single person, or groups, or organizations. Meanwhile, the victim of the hate speech can also be a single person, or groups. In [1], hate speech is defined as any communication that disparages a person, or a group based on some characteristics such as race, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics. The spread of hate speech is very dangerous in human social life, such as people thought will become more bias, more people will discriminate others, people tend to have less empathy, and it also violates human right.

The number of spread of hate speech rises in 2016. By 2017, there were 3325 cases that had been handled by the Indonesian police. It increased 44.99% from the previous year which is 1829 cases[1]. In 2016, the Indonesian police had detected that there were 180.000 users in social media accused of spreading hate speech[2]. Some of the cases were spreading hate speech to others, discriminate some groups, insult President and the police. Spreading hate speech was found not only through postings, comments in the social media, but also through video, campaign, and speech.

Manual filtering of hate speech is conducted by the police with the aids from the citizens' reports. An example is given in the following report:

> "@DivHumas_Polri tolong akun diatas agar segera direspond dan ditindaklanjuti karena menyebarkan ujaran kebencian"
>
> which in English is: ("@DivHumas_Polri help this account be immediately responded and followed up because of spreading hate speech")

From those reports, the police check the complains whether they are correct or not and decide what to do next. Such manual filtering is very time consuming and relies heavily to the people complaints. This is one of the reasons why automatic hate speech detection is necessary.

In research [2], they compared SVM and LSTM algorithm to detect hate speech on the Facebook using two datasets in Italian: the three-class dataset, *No Hate, Weak Hate,* and *Strong Hate*, and the two-class dataset, *No Hate, Hate*. In [3], they used deep learning algorithm such as LSTM, CNN and FastText with textual features GLoVe and Random Embedding to classify hate speech in English. Meanwhile in [4] and [5], they used Indonesian dataset and compared the performances of textual features with several machine learning algorithm.

In this research, we created a new dataset from several social media such as Facebook, Twitter, Youtube, and LineToday for training deep learning models to detect hate speech in Indonesian Language. We performed hate speech detection for Indonesian Language using textual and acoustic features. The reason of performing hate speech detection using acoustic features is because we encountered that spreading hate speech can also be done through video. We trained textual model, acoustic model and multi-features model using LSTM to detect hate speech.

The next section describes some related works. In Section III, we define the corpus construction of hate speech, how features extracted, and our purposed method. In Section IV, we present the details and discuss our experiments result. Finally, Section V concludes the paper.

[1] https://news.detik.com/berita/d-3790973/selama-2017-polri-tangani-3325-kasus-ujaran-kebencian

[2] https://www.bbc.com/indonesia/berita_indonesia/2015/11/151102_trensosial_ujaran_kebencian

978-1-7281-1175-9/18/$31.00 ©2018 IEEE

II. RELATED WORKS

There was research conducted by [2] using Italian dataset crowdsourcing from Facebook. The research conducted machine learning by comparing SVM and LSTM algorithm. They used morpho-syntactically tagged texts that automatically tagged by the Part-Of-Speech tagger approach. The research has two datasets which are *No Hate, Strong Hate* and *Weak Hate* dataset and *No Hate* and *Hate* dataset. The three classes dataset conducted 100% agreement and the two classes dataset conducted two experiment 100% agreement and more than 70% agreement.

In [3], they proposed the textual features of random embedding and GloVe embedding. In their research, they used CNN, LSTM and FastText as the classifier builder. The research has dataset which is racist, sexist, nor both classes. The optimizer that they used was *adam* for CNN and LSTM and *RMS-Prop* for FastText.

In Indonesia research [5], they focused on religion, race, ethnicity, and gender dataset. They used word n-grams with n=1 and n=2, character n-gram with n=3 and n=4, and negative sentiment as the textual features. They compare four model which are Naïve Bayes, Support Vector Machine, Bayesian Logistic Regression, and Random Forest Decision Tree to detect hate speech in Indonesian Language. The dataset consists of *No Hate Speech* and *Hate Speech* class. The best model went to Random Forest Decision Tree with 93.5% F-measure score. The result showed that word n-gram outperformed character n-gram.

III. PROPOSED METHODS

From the related works, we proposed our methods. We are not using the character n-gram such as in the research [5], instead that we use the word n-gram. Here, we also use deep learning approach, we use LSTM that also used in [2] and [3]. In order to do that, we created our own dataset for Indonesian Language as we could not find such dataset in Indonesia language so far. The new dataset is necessary because we want to have more varied topics, not just about political issues [5]. The following explains the detail about how the datasets constructed, data pre-processing and feature extraction.

A. Datasets Development

This section consists of two major process, data acquisition and data annotation of hate speech dataset for Indonesian language.

1) Data Acquisition

There are two types of data, audio and text. The target was 1100 hate speech and 1100 no hate speech. The data train are 1000 hate speech and no hate speech. The data test are 100 hate speech and no hate speech. Text data were collected manually from social media such as Facebook, Twitter, Line Today, YouTube comments, and YouTube video transcript. In Facebook and Twitter, search engines were used to search targeted keyword. Meanwhile, in YouTube and Line Today, hot topic features were used to get potentially hate speech comment. He/she can be politician, celebrities, students, etc. Audio data were collected through recording 19 men and 8 women by reading expressively the text data collected before. The speakers are around 19 to 21 years old. Each speaker speaks 60 to 110 texts which consists of hate speech and no hate speech. The audio data got duplicated between men and women. The results of data collection can be seen in Table I.

TABLE I. The number of collected data

Data Type	Hate Speech	No Hate Speech	Total
Text	1173	1100	2273
Audio	1226	1243	2469

2) Data Annotation

Each data on the datasets was annotated manually into two classes which are:

1. **Hate Speech**: a sentence consists of insults, defamation, blasphemy, provocation and instigation.
2. **No Hate Speech**: a sentence consists of neutrals, positive, constructive and non-offensive.

All of the collected sentences described in Table I were annotated by 3 people. The data with hate speech class were labeled as 'H' and no hate speech class were labeled as 'NH'. We gave the definition of hate speech to those annotators for reducing bias during the labeling process, including some sentences examples which consists of insults, defamation, blasphemy, provocation and instigation.

After all annotators completed the sentence labelling, the final label of each sentence is determined based on the votes of those of 3 annotators. We only assign Hate Speech label or "H" label only to sentences that have 100% agreement. This is similar for the No Hate Speech label or "NH label, we were taking 100% agreement of no hate speech data. Because of no background consideration, 100% agreement of hate speech was taken to avoid bias. No used data were discarded and immediately search for the new data until the number of hate speech data and no hate speech data meets the target amount of the data in the datasets.

B. Data Pre-processing

After the data is fully annotated, the data were preprocessed by:

- Cleaning up the data such as removing URLs, tags (#, @, etc.), and emoticon (□, □). This is because the URLs does not give any valuable information. For tags and emoticon, they can give the victims and expression information, but we are not focusing on those features in this research.
- Abbreviation such as "yg", "dpt", "jkt", "utk", "ttg" change into the standard form.
- Lowercasing all the characters and changing numbers into value.
- Removing beginning silent and end silent in audio data.

C. Features Extraction

We use textual, acoustic and the combination of textual and acoustic features to find the best feature set in hate speech detection. The textual features were word n-grams and word embedding. For word n-grams, we used unigram, bigram, trigram, and the combination between the n-grams with BOW (Bag-of-Words) and TF-IDF (Term Frequency-Inverse Document Frequency) representation. For word embedding, we implemented CBOW (Continuous Bag-of-Words) and skipgram [6]. As for the acoustic feature, we used prosody, MFCC_0_D_A, MFCC_E_D_A, INTERSPEECH 2009 [7], and INTERSPEECH 2010 [8]. Those features were extracted from audio using openSMILE.

D. Models Developments

We used deep learning model based on the Long Short-Term Memory (LSTM) topology in detecting hate speech in Indonesian language. LSTM was firstly proposed by Hochreiter and Schmidhuber [9]. This network is a modification of Recurrent Neural Network (RNN) which aims to overcome short-term memory problems. LSTM can keep information in longer time and when needed it can be reused. RNN has a layer of neural network. Meanwhile, LSTM has 4 layers of neural network that connected each other. Those layers are memory cell, input gate, output gate, and forget gate. Memory cell is a place to store the long-term information, input gate decides which information will be written to the memory, forget gate chooses what existing information to be deleted from memory, and output gate controls the amount of information out from the hidden state. We created the LSTM model using three kind of features:

1) Textual model

The model is assigned to classify sentences using textual features. The data were extracted into two types of features, word n-grams and word embedding. The word n-grams combination were unigram, bigram, trigram, uni_bigram, uni_trigram, uni_bi_trigram, bi_trigram. Meanwhile the word embedding achitecture had the size 200. The LSTM topology that we used can be seen in Figure 1. As it can be seen from Figure 1, we used 40 sequence word padding size and 200 features size for the input word embedding. Meanwhile, for the input word n-grams we used 1 sequence size and m features size (m depends on the selected n-grams). We also used 0.5 dropout and adam optimizer. The output function is Sigmoid.

2) Acoustic model

The model is assigned to classify audio using acoustic features. The lld (low-level descriptor) features were extracted by using openSMILE with default configuration from the tools. The LSTM Topology that we used can be seen in Figure 1. For acoustic model, we used the same topology as the textual model with the input 750 sequence frame padding size and n features size (n depends on the selected acoustic features).

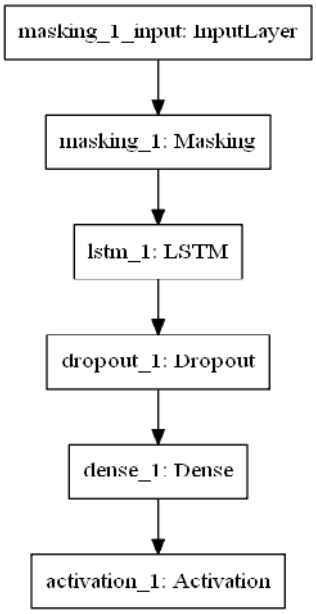

Figure 1. LSTM topology for textual and acoustic model

3) Multi-model model

The model is assigned to classify sentences and audio using both textual features and acoustic features. The LSTM topology that we used can be seen in Figure 1.

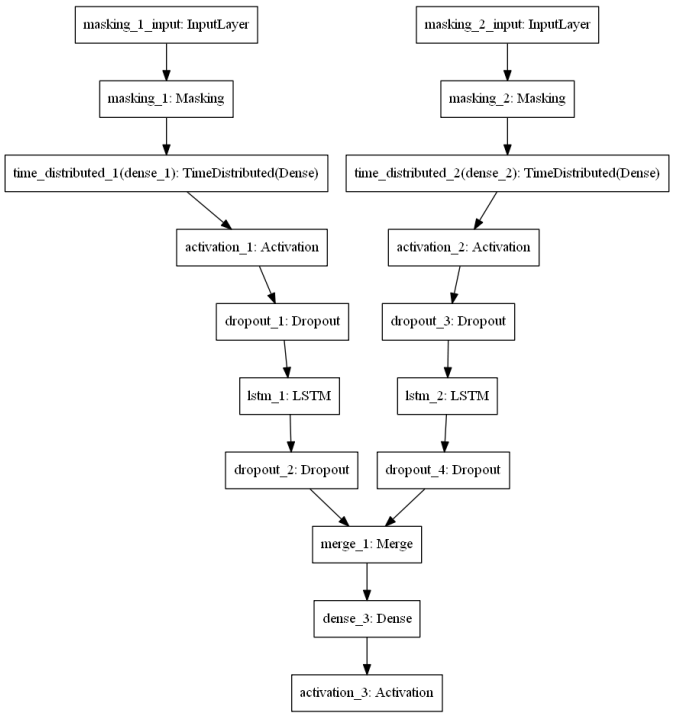

Figure 2. LSTM topology for multi-model model

As it can be seen from Figure 2, the left side is textual model with 40 sequence word padding size and 200 features size, 2 hidden layers. The right side is acoustic model with 750 sequence frame padding size and n features size (n depends on the selected acoustic feature) and 2 hidden layers. We used a 0.5 dropout

value and adam optimizer. The output function is Sigmoid.

IV. EXPERIMENTS AND RESULTS

In this section, we give the details of the experiments results and analysis.

A. Experiments Results

1) Textual model

There were two types of experiments. First, we used the word n-grams features with the combination among the word n-grams. Seconds, we used word embedding features. The result of hate speech classification using word n-grams and word embedding as follows.

TABLE II. Experiment results using word n-grams combination

Features	F1-score (%)
UNI_BOW	83.02
BI_BOW	75.29
TRI_BOW	40.37
UNI_BI_BOW	**83.91**
UNI_BI_TRI_BOW	79.65
BI_TRI_BOW	70.6
UNI_TFIDF	82.43
BI_TFIDF	74.96
TRI_TFIDF	40.47
UNI_BI_TFIDF	83.82
UNI_BI_TRI_TFIDF	82.82
BI_TRI_TFIDF	72.51

Table II shows F1-score of the classification result for each combination features of word n-grams. Features using the combination between unigram and bigram with BOW representation has the best F1-score (83.91%). Using only trigram with BOW representation and TF-IDF representation are not recommended because it has the worst F1-scores among the word n-grams. But, the combination between unigram or bigram with trigram still recommended.

TABLE III. Experiment results using word embedding

Features	F1-score (%)
Skipgram	56.91
CBOW	**87.98**

Table III shows F1-score of the classification result for word embedding. Features using CBOW has better F1-score (87.98%) than skipgram. Among the textual features, CBOW has the best F1-score.

2) Acoustic model

The result of the hate speech classification using acoustic features can be seen in Table IV. The result shows that using MFCC_E_D_A features has the best performance (72.49%) among the other four acoustic features. The second and the third best from the result which is INTERSPEECH 2010 (72.43%) and MFCC_0_D_A (72.12%) have not much different from MFCC_E_D_A F1-score. Compared to the model using textual features, acoustic features were outperformed by the model using textual features.

TABLE IV. Experiment results using acoustic features

Features	F1-score (%)
Prosody_Acf	60.78
MFCC_0_D_A	72.12
MFCC_E_D_A	**72.49**
INTERSPEECH 2009	66.43
INTERSPEECH 2010	72.43

3) Multi-features model

The result of the hate speech classification using multi-features can be seen in Table V. The result shows that using the combination between word embedding and INTERSPEECH 2009 has the best performance among the other four combination features. Compared to textual model, multi-model's best result has not much different from the textual model's result. But, as we can see, the multi-model result has average above 83%.

TABLE V. Experiment results using multi-model model

Features	F1-score (%)
WE_prosody_acf	86.3
WE_mfcc0	83.8
WE_mfcce	86
WE_IS09	**86.98**
WE_IS10	83.84

B. Discussion

Based on the experiment results, the textual model can detect hate speech more accurate than that of the acoustic model and that of the multi-features model. Among the features we used, word embedding with CBOW architecture had the best result followed by the combination between CBOW and INTERSPEECH 2009. The use of acoustic features was not enough to detect hate speech. At least, to improve the result of using acoustic features, the textual features are necessary.

We found the model using word embedding with the CBOW (87.98%) was better than using word n-gram and their combination (the highest achieved 83.91%). We decided to experiment the combination of CBOW and all the acoustic features that we used in Experiment Table IV. The result (86.98%) was not much different from the textual model best result (87.98%). By using the same dataset and the brute force method such as string matching based on keyword, this model shows that it outperformed that method.

We analyzed the classification results carefully. We found that there are several incorrect results due to the bias of some people. Table VI shows the example of the incorrect classification due to such case. From the examples, we figured out that the need of changing person name into "PERSON" or else due to the data did not cover all the person name. Hence, the classifier identified such sentences incorrectly.

TABLE VI. The example of the wrong classification due to name

True Class	Sentences	Prediction Class
HS	Ruhut penjilat. Jilat aja terus bang Ruhut (Ruhut is bootlicker. Just keep praising, bro Ruhut.)	HS
HS	Budi penjilat. Jilat aja terus bang Budi (Budi is handshaker. Just keep praising, bro Budi.)	NH

Besides this mistake, there were also other example of the wrong classification. We found out that these several structure sentences such as "Bongkar saja + O" (), "Tenggelamkan" (sink), "Bubarkan" (disband), "Musnahkan + O" (destroy + O) lead to wrong classification. Table VII shows the example of wrong classification.

TABLE VII. The example of wrong classification

True Class	Sentences	Prediction Class
NH	Bongkar saja dulu rumahnya, nanti baru dibangun lagi (Just demolish the house, then it will be built again)	HS
NH	Gak makan ikan, saya tenggelamkan. suka jargonnya, Bu. Keren banget! (not eat fish, I will sink it. Like the motto, miss. It's so cool!)	HS
NH	rasanya kita sudah tidak waras lagi kalau menyetujui kpk dibubarkan (feel like we aren't sane anymore if we agree it to be disbanded.)	HS
NH	Musnahkan kertas itu, sudah tidak berguna lagi (throw the paper away, we don't use anymore.)	HS

Furthermore, we also found that the preprocessing of dataset still need improvement. For example, there are still several words that have same meaning, such as "ga ngerti", "gak ngerti", "tidak mengerti" which means do not understand. The other reason was the number of out-of-vocabulary (OOV) were increasing due to the several words that have same meaning.

The use of acoustic features did not outperform both the textual features and the combination of textual and acoustic features. The best acoustic model has 72.49% by using MFCC_E_D_A configuration (MFCC with log energy). The close result of acoustic model has 72.43% by using INTERSPEECH 2010 features. The lowest acoustic model has 60.78% by using prosodic auto-correlation (prosody_acf) features.

V. CONCLUSION

In this paper, we introduced hate speech detection in Indonesian Language using deep learning approach based on the LSTM by employing textual features, acoustic features and the combination of both features. Among the constructed model, textual model had the highest performance followed by the multi-model model. The acoustic model was outperformed by the textual model and multi-model. The best textual model that we used was word embedding with CBOW architecture. It achieved F1-score 87.98% in detecting hate speech.

VI. FUTURE WORK

We suggested that the word normalization was conducted in the upcoming research to decrease the OOV and the same meaning word/phrase as the example in the discussion part. Also, by changing the person name or pronoun into "PERSON" or else could also be experimented to check whether the performance will increase or not. We also suggested the use another word embedding architecture such as FastText that has been conducted by [3]. Last, we suggested using normalization on the audio because we had not tried to normalize the audio.

VII. ACKNOWLEDGEMENT

This research is partially funded by PENELITIAN TERAPAN UNGGULAN PERGURUAN TINGGI program with title '*Sistem Cerdas Pemantau Perilaku Penggunaan Gadget di Kalangan Remaja Menggunakan Teknik Pembelajaran Mesin*' (intelligent system for monitoring gadget usage behavior among teenagers using machine learning). We also thank PT. Prosa Solusi Cerdas and other parties who contributed in this experiment or during the process of working for this paper.

REFERENCES

[1] W. Warner and J. Hirschberg, "Detecting Hate Speech on the World Wide Web," in *Proceedings of the 2012 Workshop on Language in Social Media*, Montreal, Canada, 2012.

[2] F. D. Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi and M. Tesconi, "Hate Me, Hate Me Not: Hate Speech Detection on Facebook," in *In Proceedings of the First Italian Conference on Cybersecurity*, Venice, 2017.

[3] P. Badjatiya, S. Gupta, M. Gupta and V. Varma, "Deep Learning for Hate Speech Detection in Tweets," in *International World Wide Web Conference Committee*, Perth, 2017.

[4] S. H. Partiwi, "Detection of Hate Speech against Religion on Tweet in the Indonesian Language Using Naive Bayes Algorithm and Support Vector Machine," 2016.

[5] I. Alfina, M. I. Fanany, R. Mulia and Y. Ekanata, "Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study," in *9th International Conference on Advanced Computer Science and Information Systems*, 2017.

[6] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representation in Vector Space," 2013a.

[7] B. Schuller, S. Steidl and A. Batliner, "The Interspeech 2009 Emotion Challenge," in *10th Annual Conference of the International Speech Communication Association*, 2009.

[8] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller and S. Narayanan, "The INTERSPEECH 2010 paralinguistic challenge (2010)," in *In Proc. Interspeech*, 2010.

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," 1997.