Synonym based feature expansion for Indonesian hate speech detection

Imam Ghozali, Kelly Rossa Sungkono, Riyanarto Sarno, Rachmad Abdullah
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Feb 20, 2022
Revised Sep 2, 2022
Accepted Sep 18, 2022

Keywords:

Bidirectional long short-term memory
FastText
Hate speech
Synonym
Word2Vec

ABSTRACT

Online hate speech is one of the negative impacts of internet-based social media development. Hate speech occurs due to a lack of public understanding of criticism and hate speech. The Indonesian government has regulations regarding hate speech, and most of the existing research about hate speech only focuses on feature extraction and classification methods. Therefore, this paper proposes methods to identify hate speech before a crime occurs. This paper presents an approach to detect hate speech by expanding synonyms in word embedding and shows the classification comparison result between Word2Vec and FastText with bidirectional long short-term memory which are processed using synonym expanding process and without it. The goal is to classify hate speech and non-hate speech. The best accuracy result without the synonym expanding process is 0.90, and the expanding synonym process is 0.93.

Corresponding Author:

Riyanarto Sarno
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
Email: riyanarto@if.its.ac.id

1. INTRODUCTION

Online hate speech is one of the negative impacts of the development of internet-based social media. Hate speech is a kind of communication that is rude, abusive, intimidating, harassing towards a person or group of people [1]. It can cause a detrimental impact directly or indirectly to other people or groups [2]. An example of the hate speech spread is during the regional head election or presidential election, in which each group attacks one another in order to win the election results.

Hate speech can happen because people have lack of understanding to differentiate between criticism and hate speech. They often use dirty words, harassment, and even insult [3]. The Indonesian government published a regulation regarding hate speech in *Undang-Undang Informasi dan Transaksi Elektronik* (Law of Electronic Information and Transaction), article 27 paragraph (3) and article 45 paragraph (1), and the circular (SE) of the National Police Chief number SE/6/X/2015 [3]. Online hate speech is also a problem in other countries. For example, in Rohingya Myanmar, in 2017, online hate speech incited violence against Rohingya Muslims and killed thousands of civilians [4]. Twitter is one of the social media which is potentially used to express hate speech, and people can freely tweet their feelings, opinions, or criticisms of a government policy. Therefore, it is necessary to detect hate speech before a crime occurs.

Hate speech detection in Twitter data includes natural language processing (NLP) text classification. However, most of the existing research about hate speech only focuses on feature extraction and classification methods [3], [5]–[8]. For example, research aiming to detect hate speech in English was conducted by combining the four feature extraction methods [5]. These features are sentiment-based, semantic features, unigram features, and pattern features. After getting features, machine learning was used for classification. D'Sa *et al.* [9] detected toxic speech with a deep learning approach, while Fauzi and Yuniarti [7] used an ensemble technique to classify Indonesian hate speech.

In natural language processing (NLP), feature extraction plays an essential role in getting good results because its results are used in classification methods. Word2Vec is a feature extraction that is not better than term frequency-inverse document frequency (TF-IDF) and term frequency (TF). TF-IDF and TF got better results when combined with machine learning algorithms such as support vector machine (SVM), naïve Bayes (NB), Bayesian logistic regression (BLR), and random forest (RF) as classifiers to recognize hate speech in the Indonesian language [2]. Expanding features with synonyms, hypernyms, and holonyms can improve accuracy [10].

This paper used synonyms-based feature expansion on word embedding methods Word2Vec and FastText, then applied bidirectional long short-term memory (BiLSTM) for classification to produce high accuracy. The rest of the paper is constructed in the other four sections. Related work is reported in Section 2. Section 3 presents our proposed methods. Experimental setup and result analysis are explained in section 4, while section 5 contains the conclusion of our paper with future scope.

2. RELATED WORK

2.1. Hate speech detection

Many methods have been proposed in hate speech detection. Detecting hate speech on Instagram had been proposed using combination method of Word2Vec Skip-gram model, random oversampling method, modification of TextCNN to get a high F1-score on an imbalanced dataset [6]. In the research, oversampling and undersampling methods are also used to handle imbalanced dataset.

Detecting hate speech by combining four features extraction with machine learning classifier obtain 0.874 accuracy for two classes (offensive and non-offensive) and 0.784 accuracy for three classes (hateful, offensive, and clean) in English dataset [5]. When convolutional neural networks (CNN) and TF-IDF are used to identify hate speech in the Indonesian tweets, the best accuracy obtained from the test was 0.825 and 0.73 for the lowest [8]. Since this research used word frequencies, using word sequences feature extraction Word2Vec or FastText is suggested to improve accuracy. An ensemble technique is used to classify Indonesian hate speech by combining machine learning such as naïve Bayes (NB), k-nearest neighbors (KNN), maximum entropy (ME), RF, and SVM [7]. Using soft voting and hard voting methods to combine these machine learning methods, the research showed that the ensemble approach can minimize misclassification when detecting new data.

Gated recurrent unit (GRU) is used for classification and Word2Vec for feature extraction [2]. GRU is a variant of recurrent neural network (RNN), and it was selected because it has simpler architecture than long short-term memory (LSTM). Besides, just like LSTM, GRU also solves the vanishing gradient problem in standard RNN. GRU is also used for hate speech detection [2], but the FastText pre-trained model for feature extraction is used without pre-processing step. The pre-trained model from Wikipedia obtained the best results. Table 1 shows the comparison of data and methods from previous research.

Table 1. The comparison data and methods

No	Reference	Data	Methods
1	[5]	– 21,000 tweets for training – 2,010 tweets for testing 2,010 tweets for validation	Sentiment-based features, semantic features, unigram features, pattern features and J48graft algorithm
2	[7]	– 260 labelled as hate speech 445 labelled as non-hate speech	TF-IDF, NB, KNN, ME, RF, SVM, hard voting and soft voting
3	[2]	– 260 labelled as hate speech 453 labelled as non-hate speech	Word2Vec and GRU
4	[6]	– 1,018 labelled as hate speech – 12,176 labelled as non-hate speech 80% for training and 20 % for testing	Word2Vec Skip-gram model, TextCNN, synthetic minority oversampling technique (SMOTE) and TextCNN +LSTM
5	[8]	– 630 tweets for training 5, 50, 75, 100, 150, and 200 tweets for testing	TF-IDF and CNN
6	[3]	– 260 labelled as hate speech – 453 labelled as non-hate speech 80% for training and 20% for testing	FastText and GRU

2.2. Synonym-based feature expansion

Fauzi *et al.* [11] studied the sentiment analysis on the mobile banking review app, which then obtained the fact that adding synonym-based feature expansion was better than those that did not use feature expansion in the naive bayes classifier. Adding WordNet relations, such as synonyms, hypernyms, and holonyms, into feature extraction can increase the accuracy of the classifier [10]. Comparing synonym candidates with parents, children, or root nodes in large shopping taxonomies increased the accuracy of classification included in the complex entities [12]. WordNet is used to get synonyms and antonyms for expanding the meaning of words [13]. Word list with term frequency-inverse cluster frequency (TF-ICF) was extended to overcome the out-of-vocabulary (OOV) problems [14].

2.3. Word embedding

Word embedding converts words to real vector numbers. Word embedding can capture semantic similarity and describe it into a small vector dimension, which differs from the one-hot encoding approach. Each word is represented as an integer number which makes the vector representation very large [15]. Based on previous research [16], we used two word-embedding Word2Vec and FastText because these methods could get great results for binary classification.

2.3.1. Word2Vec

Word2Vec is a method proposed as a model for effective training on large documents with low computational complexity [15]. It can be used to solve problems in NLP. Word2Vec and deep learning CNN, LSTM, and CLSTM is used for traffic events detection, where Word2Vec and CNN got the best results for binary and ternary classification compared to the FastText model and random word vectors [16]. The implementation of Word2Vec and CNN to classify short sentences product reviews explained that Word2Vec and CNN were able to do text classification [17]. The use of LSTM is suggested to handle sequential data in text classification in future work.

Word2Vec has two types: continuous bag of word (CBOW) and skip-gram. CBOW is used to predict words based on context by maximizing word probability and Skip-gram for predicting context based on terms because Skip-gram used the surrounding word in sentences for prediction [18]. The CBOW and Skip-gram architecture has three layers. The first layer is the input, followed by the hidden layer, and the last layer is the output. In Skip-gram, the input layer received vector data the same as the input word. The output layer provided a probability distribution for all words in the context.

2.3.2. FastText

FastText is an expanded version of the Word2Vec skip-gram model [19]. It uses the Word2Vec skip-gram model designed to study the vectors which is represented by each n-gram character in the text corpus. Then, the vector representing the word is the sum of the vectors associated with the n-gram vector [16]. In other words, FastText creates vectors by learning the information subword [20]. For example, with *n*=3, the word <*penjara*> (<jail>) can be represented. <*pen, enj, nja, jar, ara*> (<jai,ail>). The vector of the word *penjara* (jail) is the sum of all the n-grams that appear. The word w indicates that $N_w \in \{1, \ldots., N\}$ is part of the n-grams that appears in the word w.

2.4. Text classification

Recently the deep learning approach is often used to complete text classification tasks. CNN and RNN are two methods that solve text classification problems [17], [21], [22]. However, both have disadvantages. CNN cannot handle sequential data directly, that using word embedding to produce an optimum performance for sentiment text classification is suggested [23]. RNN, having different problems, sums all the backpropagation phases that can make the gradient smaller or larger, which is commonly known as vanishing or exploding gradient [24]. Due to those drawbacks, LSTM was introduced [25] to solve the problems and became the development of RNN.

LSTM is an extension of RNN that can solve the gradient vanishing or exploding problems on RNN [24]. Vanishing gradient is caused by the gradient which is getting smaller until the last layer. It then makes the weight value does not change so that it never gets better or convergent results. LSTM consists of three gates and a cell memory state. Although LSTM can handle many text classification problems, LSTM only uses the results from previous information, and sometimes it is not enough to solve classification problems [26]. Therefore, BiLSTM was developed. The BiLSTM consists of two LSTMs, which make it possible to get information from both forward and backward directions and combine the knowledge for a better result. The equations of BiLSTM output can be seen in (1) [27],

$$y_k(t) = \overrightarrow{y_{\mathrm{k}}}(t) + \overleftarrow{y_k}(t) \tag{1}$$

where y_k represents the result of BiLSTM, $\overrightarrow{y_k}$(t) isthe ouput of the forward LSTM, and $\overleftarrow{y_k}(t)$ is the output of the reserve LSTM.

2.5. Evaluation

Precision, recall, and F1-score are used to measure the performance evaluation of our model [2]. Precision divides the positive true class predictions and the overall positive class predictions. Then recall is a true positive division by the total TP and FN. Finally, F1-score is the harmonic mean of precision and recall.

3. METHOD

The proposed method diagram is shown in Figure 1. There are two stages in our research regarding hate speech detection. The first is the training stage and the second is the testing stage. In the training stage, we pre-processed data to clean the data before entering the synonym feature expansion. Then results of the synonym feature expansion were used for feature extraction. The following process was a classification to make a model. The testing phase served to test the model that had been formed in the training phase. First, we also pre-process data test before the detection process. Then, our hate speech detection results were presented in binary classification: hate speech or non-hate speech.

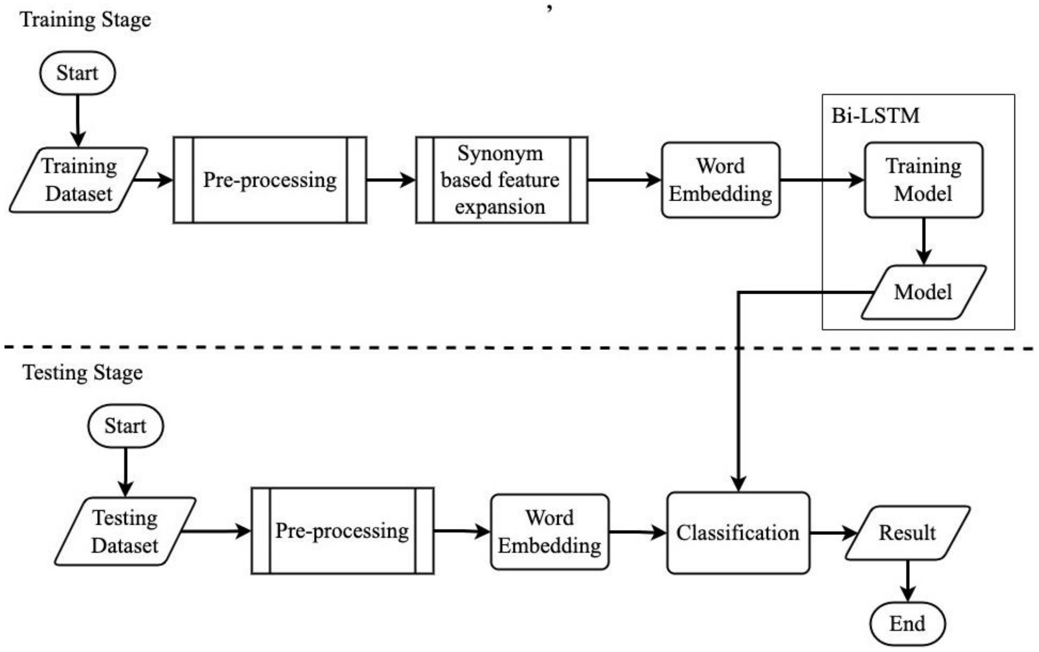

Figure 1. The proposed method

3.1. Data collection

We used a dataset from Alfina *et al.* [28], which contains data from Indonesian Twitter. The dataset consists of 713 data which contains 260 labeled hate speech and 453 non-hate speech. It was collected in 2017 and related to political events, namely the 2017 Jakarta gubernatorial election. The dataset was annotated by 30 annotators consisting of college students studying in Jakarta and the surrounding area at the age of 17-24 years old. An example of a dataset can be seen in Table 2.

Table 2. The example of dataset [28]

No	Text	Label
1	*Tuhan punya rencana yg indah buat Pak Ahok dan Pak Djarot* (God has beautiful plan for Mr. Ahok and Mr. Djarot)	Non_HS
2	*Shame on you silvy!! Ga malu fitnah?? Tuh rasain pak ahok panas, skak mat #DebatFinalPilkadaJKT* (Shame on you silvy!! Aren't you ashamed of being a slander?? See, Pak Ahok is angry, checkmate *#DebatFinalPilkadaJKT*)	HS

3.2. Pre-processing

This stage was aimed to make cleaner data before entering the next step, this stage consists of five steps: the first is case folding stage to change words into lower case, then filtering function to remove special characters so that only letters and numbers are left, stop word removal process to remove meaningless words which are (and, or, to, and from), stemming function to change words into original words, tokenizing is a process to separate words in a sentence into tokens.

3.3. Synonym-based feature expansion

This process was done to expand features by adding the identical meaning of a word to make the number of features more extensive. In this study, Indonesian thesaurus [29] was used to get synonyms for each word. The first step in this process was to get the word from the pre-processing results. The second step was to find the synonym from the Indonesian thesaurus. In the last step, we used 100% results synonym from the thesaurus and inserted as the new features. The example of synonym-based feature expansion is shown in Table 3. The bold word represents the original word, and the non-bold word represents the synonym expansion.

Table 3. The example of synonym expansion [29]

No	Process	Results
1	Original tweet	*Tuhan punya rencana yg indah buat Pak Ahok dan Pak Djarot* (God has beautiful plan for Mr. Ahok and Mr. Djarot)
2	Pre-processing	[*'tuhan', 'rencana', 'indah', 'ahok', 'djarot'*] ([god, plan, beautiful, ahok, djarot])
3	Synonym expansion	[***tuhan***, ***rencana***, *acara, agenda, bagan, buram, dasar, draf, jadwal, konsep, persediaan, persiapan, program, rancangan, rangka,* ***indah***, *artistik, bagus, baik, bergaya, cakap, cantik, dikara, elok, jalak, minat,* ***ahok***, ***djarot***] ([**god**, **plan**, program, agenda, schematic, sketch, base, draft, schedule, supply, preparation, programme, design, framework, **beautiful**, artistic, good, nice, stylish, capable, pretty, noble, elegant, starling, interest, **ahok, djarot**])

3.4. Word2vec

This process converts the results of the synonym-based feature expansion into a vector value. We used Word2Vec for word embedding skip-gram. We also used Gensim library in Python to implement Word2Vec with vector size 100, windows size 5, and minimum count 3, then the result of vectors to train and test with classification methods. In this process, we trained the model with the original word from the Indonesian tweet and added 100% synonym expansion from the thesaurus.

3.5. FastText

This process changed the results of synonym-based feature expansion using the FastText method. This study used the Gensim library in Python with vector size 100, windows size 5, and minimum count 3. Same as Word2Vec in this process, we trained the model with the original word from the Indonesian tweet and added 100% synonym expansion from the thesaurus.

3.6. Classification

This stage was divided into two: training and testing phase. The training data, which had passed the preprocessing phase, the synonym-based feature expansion stage, and the word embedding stage, entered the training phase. The word embedding results were trained with BiLSTM with predetermined parameters to create a model. The model from the training phase was used to classify testing data in the testing phase. The results of preprocessing and word embedding in the testing phase was a model. The prediction model from the testing phase was then compared to the model from the training phase. If the prediction model was less than 0.5, the text data would be classified as hate speech. Meanwhile, if it was more than or equal to 0.5, the text would be classified as non-hate speech. The example of the classification result can be seen in Table 4.

Table 4. The example of dataset [28] and the classification result

No	Text	Prediction value	Predicted class	Actual class
1	*Tuhan punya rencana yg indah buat Pak Ahok dan Pak Djarot* (God has beautiful plan for Mr. Ahok and Mr. Djarot)	0.798	Non_HS	Non_HS
2	*Shame on you silvy!! Ga malu fitnah?? Tuh rasain pak ahok panas, skak mat #DebatFinalPilkadaJKT* (Shame on you silvy!! Aren't you ashamed of slander?? See, Pak Ahok is angry, checkmate#DebatFinalPilkadaJKT)	0.252	HS	HS

4. RESULTS AND DISCUSSION

In this section, the performance of our study is described. First, the method to compare Word2Vec and FastText with the BiLSTM classifier is explained. Then, Word2Vec and FastText are also compared, after using synonym-based feature expansion. In the last part of this section, the results are compared with other methods.

4.1. Model performance

In measuring model performance, four models are compared: Word2Vec, "Word2Vec+synonym-based feature expansion", FastText, "FastText+synonym-based feature expansion" with 20% data from dataset or 143 data for test. In this experiment, BiLSTM is used for classification with 10, 20, 30, 40 and 50 epochs. The average obtained by FastText was better than Word2Vec and the result of adding synonym-based feature expansion gave higher accuracy. The best average of "FastText+synonym-based feature expansion" obtained was 0.88112, and the highest accuracy result was 0.9301. Table 5 represents the average accuracy of models.

Table 5. Average accuracy of models

Model	Average accuracy
Word2Vec	0.857
Word2Vec+synonym	0.869
FastText	0.871
FastText+synonym	0.881

4.2. Comparison results

In previous research, the best F1-score with NB, SVM, and RF was got when combined with hard or soft voting [7]. The results obtained by hard and soft voting were the same, which was 0.798. GRU was also used to detect hate speech [2], [3], which obtained better results. Even though GRU was used, the feature extraction was different. The use of Word2Vec [2] obtained better result than the use of FastText model [3] in the precision and F1-score. The training data influenced the Word2Vec method. With more training data, the Word2Vec model has a greater chance to represent the suitable word. FastText has better capabilities because it can handle OOV problems.

We compared our results with the previous study that used the same dataset [28] and got better precision and accuracy results. However, the recall and F1-score obtained were not better than previous research because the recall and the F1-score result on the proposed method model were low to recognize the hate speech class. The recall result was 0.8444, while the non-hate speech class reached 0.9694, and the F1-score reached 0.8837 for hate and 0.9500 for non-hate. With imbalanced datasets, some data were misclassified. The example "*Sylvi terlihat bloonnya #DebatFinalPilkadaJKT* (Sylvi looks stupid *#DebatFinalPilkadaJKT*)" was classified as non-hate speech. In previous research, Patihullah and Winarko [2] got the best accuracy with 0.9296, and the proposed method reached 0.9301.

We used BiLSTM for the classifier. Although the BiLSTM architecture is more complex than the GRU, BiLSTM gets information in forward and backward directions, combines the knowledge, and gets a better result. Synonym-based feature expansion was added to the feature extraction. The previous test showed that FastText+synonym-based feature expansion got the best average than the other models. Table 6 shows its comparison with other methods.

Table 6. Comparison results

No	Method	Result			
		Precision	Recall	F1-score	Accuracy
1	The Proposed Method	**0.930**	0.907	0.917	**0.930**
2	[2]	0.885	**0.920**	0.902	0.930
3	[3]	0.923	0.915	**0.919**	-
4	[7]	-	-	0.798	-

5. CONCLUSION

This study proposes to add synonym-based feature expansion at word embedding to recognize hate speech on Indonesian's Twitter. We compared the performance of Word2Vec and FastText to convert words into vectors. We also used the BiLSTM as the classifier. Synonym-based feature expansion in word embedding could impact the average accuracy. The best result in our study with an accuracy of 0.9301 was obtained using FastText+synonym-based feature expansion and got the best effects in the average accuracy test. Word2Vec has increased accuracy by 0.0112, and FastText 0.0098 compared to the one without synonym-based feature expansion. For future work, we suggest using antonyms, hypernyms, and holonyms for expanding features and using transformer models as bidirectional encoder representations (BERT).

ACKNOWLEDGEMENTS l

This research was funded by the Indonesian Ministry of Education and Culture, and the Indonesian Ministry of Research and Technology/National Agency for Research and Innovation, under Penelitian Dana Departemen managed by Institut Teknologi Sepuluh Nopember (ITS) research grant (Contract No. 1745/PKS/ITS/2022).

REFERENCES

[1] K. Erjavec and M. P. Kovačič, "'You don't understand, this is a new war!' analysis of hate speech in news web sites' comments," *Mass Communication and Society*, vol. 15, no. 6, pp. 899–920, Nov. 2012, doi: 10.1080/15205436.2011.619679.

[2] J. Patihullah and E. Winarko, "Hate speech detection for Indonesia tweets using word embedding and gated recurrent unit," *Indonesian Journal of Computing and Cybernetics Systems (IJCCS)*, vol. 13, no. 1, Jan. 2019, doi: 10.22146/ijccs.40125.

[3] G. B. Herwanto, A. M. Ningtyas, I. G. Mujiyatna, K. E. Nugraha, and I. N. Prayana Trisna, "Hate speech detection in Indonesian twitter using contextual embedding approach," *Indonesian Journal of Computing and Cybernetics Systems (IJCCS)*, vol. 15, no. 2, Apr. 2021, doi: 10.22146/ijccs.64916.

[4] E. W. Pamungkas, V. Basile, and V. Patti, "A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection," *Information Processing and Management*, vol. 58, no. 4, Jul. 2021, doi: 10.1016/j.ipm.2021.102544.

[5] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on twitter: a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE Access*, vol. 6, pp. 13825–13835, 2018, doi: 10.1109/ACCESS.2018.2806394.

[6] I. G. M. Putra and D. Nurjanah, "Hate speech detection in Indonesian language Instagram," in *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2020, pp. 413–420, doi: 10.1109/ICACSIS51025.2020.9263084.

[7] M. A. Fauzi and A. Yuniarti, "Ensemble method for Indonesian twitter hate speech detection," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 11, no. 1, pp. 294–299, Jul. 2018, doi: 10.11591/ijeecs.v11.i1.pp294-299.

[8] D. A. N. Taradhita and I. K. G. D. Putra, "Hate speech classification in Indonesian language tweets by using convolutional neural network," *Journal of ICT Research and Applications*, vol. 14, no. 3, pp. 225–239, Feb. 2021, doi: 10.5614/itbj.ict.res.appl.2021.14.3.2.

[9] A. G. D'Sa, I. Illina, and D. Fohr, "BERT and fastText embeddings for automatic detection of toxic speech," in *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, Feb. 2020, pp. 1–5, doi: 10.1109/OCTA49274.2020.9151853.

[10] T. N. Mansuy and R. J. Hilderman, "Evaluating WordNet features in text classification models," in *FLAIRS Conference*, 2006, pp. 568–573.

[11] M. A. Fauzi, R. F. Nur Firmansyah, and T. Afirianto, "Improving sentiment analysis of short informal Indonesian product reviews using synonym based feature expansion," *Telecommunication Computing Electronics and Control (TELKOMNIKA)*, vol. 16, no. 3, pp. 1345–1350, Jun. 2018, doi: 10.12928/telkomnika.v16i3.7751.

[12] A. Boteanu, A. Kiezun, and S. Artzi, "Synonym expansion for large shopping taxonomies," *Automated Knowledge Base Construction (AKBC)*, 2018, pp. 1–18. doi: 10.24432/C5PP4H.

[13] R. Abdullah, S. Suhariyanto, and R. Sarno, "Aspect based sentiment analysis for explicit and implicit aspects in restaurant review using grammatical rules, hybrid approach, and SentiCircle," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 294–305, Oct. 2021, doi: 10.22266/ijies2021.1031.27.

[14] R. Priyantina and R. Sarno, "Sentiment analysis of hotel reviews using latent Dirichlet allocation, semantic similarity and LSTM," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 4, pp. 142–155, Aug. 2019, doi: 10.22266/ijies2019.0831.14.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[16] S. Dabiri and K. Heaslip, "Developing a Twitter-based traffic event detection model using deep learning architectures," *Expert Systems with Applications*, vol. 118, pp. 425–439, Mar. 2019, doi: 10.1016/j.eswa.2018.10.017.

[17] T. Lei, R. Barzilay, and T. Jaakkola, "Molding CNNs for text: non-linear, non-consecutive convolutions," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1565–1575, doi: 10.18653/v1/D15-1180.

[18] F. Husein Wattiheluw and R. Sarno, "Developing word sense disambiguation corpuses using Word2vec and Wu palmer for disambiguation," in *2018 International Seminar on Application for Technology of Information and Communication*, Sep. 2018, pp. 244–248, doi: 10.1109/ISEMANTIC.2018.8549843.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, Dec. 2017, doi: 10.1162/tacl_a_00051.

[20] N. Pribadi, R. Sarno, A. Ahmadiyah, and K. Sungkono, "Semantic recommender system based on semantic similarity using FastText and word mover's distance," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 2, pp. 377–385, Apr. 2021, doi: 10.22266/ijies2021.0430.34.

[21] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1422–1432, doi: 10.18653/v1/D15-1167.

[22] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751, doi: 10.3115/v1/D14-1181.

[23] P. Ce and B. Tie, "An analysis method for interpretability of CNN text classification model," *Future Internet*, vol. 12, no. 12, Dec. 2020, doi: 10.3390/fi12120228.

[24] F. Ali, A. Ali, M. Imran, R. A. Naqvi, M. H. Siddiqi, and K.-S. Kwak, "Traffic accident detection and condition analysis based on social networking data," *Accident Analysis and Prevention*, vol. 151, Mar. 2021, doi: 10.1016/j.aap.2021.105973.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[26] S. Hao, J. Long, and Y. Yang, "BL-IDS: detecting web attacks using Bi-LSTM model based on deep learning," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer International Publishing, 2019, pp. 551–563.

[27] L. Wang, X. Bai, R. Xue, and F. Zhou, "Few-shot SAR automatic target recognition based on Conv-BiLSTM prototypical network," *Neurocomputing*, vol. 443, pp. 235–246, Jul. 2021, doi: 10.1016/j.neucom.2021.03.037.

[28] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, "Hate speech detection in the Indonesian language: A dataset and preliminary study," in *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2017, pp. 233–238, doi: 10.1109/ICACSIS.2017.8355039.

[29] A. L. Victoria, "Tesaurus," *GitHub, Inc.* 2021, Accessed: Nov. 20, 2021. [Online]. Available: https://github.com/victoriasovereigne/tesaurus

BIOGRAPHIES OF AUTHORS

Imam Ghozali was born in Surabaya in 1996. He received a bachelor's degree in Computer Science from the Department of Informatics, Universitas Brawijaya in 2017. He is currently pursuing the master's degree in Informatics Engineering from the Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember (ITS). His research interests include text mining and computer vision. He can be contacted at imam.ghz17@gmail.com.

Kelly Rossa Sungkono was born in Surabaya, in June 1994. She received the master's degree in computer engineering, in 2016. She is currently a Lecturer at the Institut Teknologi Sepuluh Nopember (ITS). Her research interests include databases, information systems, and machine learning. She can be contacted at kelsungkono@gmail.com.

Riyanarto Sarno is a professor at Informatics Department, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. He received the bachelor's degree in Electrical Engineering from Institut Teknologi Bandung, Bandung, Indonesia in 1987. He received M.Sc. and Ph.D. in Computer Science from the University of Brunswick Canada in 1988 and 1992, respectively. In 2003 he was promoted to full professor. His teaching and research interests include Internet of things, process aware information systems, intelligent systems, and smart grids. He can be contacted at riyanarto@if.its.ac.id.

Rachmad Abdullah received the M.M.T. degree in Information Technology Management from Institut Teknologi Sepuluh Nopember (ITS). He is currently pursuing the Ph.D. degree in informatics engineering from the Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember (ITS). His research interests include natural language processing, focusing on text mining, sentiment analysis, extraction of 3D brain MRI, machine learning, machine translation, deep learning, and Internet of things (IoT). He can be contacted at rabdullah1506@gmail.com.