ICIC Express Letters
Part B: Applications
Volume **13**, Number **6**, June **2022**

ICIC International ©2022 ISSN 2185-2766
pp. **655–662**

CYBERBULLYING DETECTION USING WORD EMBEDDING FAST TEXT

Cindy Rahayu[1], Henry Lucky[2] and Derwin Suhartono[2]

[1]Computer Science Department, BINUS Graduate Program – Master of Computer Science
[2]Computer Science Department, School of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ cindy.rahayu; henry.lucky }@binus.ac.id; dsuhartono@binus.edu

Received August 2021; accepted November 2021

Abstract. *Cyberbullying is a threat to the mental health of young people that are growing along with advances in technology. In addition, social media can also have a negative impact by posting cruel writings or making arbitrary comments without thinking about the consequences on others. This is what makes one of the occurrences of violence in cyberspace which is often called cyberbullying. Various methods have been proposed to detect cyberbullying, one of which is machine learning technology, and many studies have been carried out to overcome this problem. However, cyberbullying detection solutions still face challenges, such as cyberbullying datasets which are still difficult to obtain and have low accuracy. The purpose of this study is to build a cyberbullying detection model with optimal accuracy in identifying cyberbully conversations using word embedding. This study introduces the use of the recently released FastText word embedding as a repetition of the word to perform sentiment analysis and cyberbullying tasks. Our proposed model uses FastText word embedding. Based on the experimental results using the "Formspring.me" dataset from Kaggle and classification using SVM, we successfully obtained a recall value of 91%, precision 93%, and F1-Score 93%.*

Keywords: Cyberbullying, Word embeddings, FastText, Machine learning

1. **Introduction.** The rapid development of social media among teenagers as a communication tool that is easy to use, equipped with applications supported by Internet facilities, and can be accessed anywhere has created a major phenomenon in the flow of information. The growth of social media has also brought a new phenomenon in society as an arena for bullying behavior. Cyberbullying is an act to give negative comments that have a repetitive quality [1]. Social media provides users with a good platform for communication and information sharing and allows them to access the latest news easily. However, these platforms are also places where users are victims of bullying, bullies, or bystanders. Although most parents reported that bullying occurred in school [2], 19.2% of people said that bullying occurred through social media sites and apps. Another 11% said they had been bullied through text messages, while 7.9% believed video games were the source. At the same time, 6.8% of people on non-social media sites reported bullying, while 3.3% reported bullying via email [2].

Along with the development of natural language processing (NLP) in machine learning, several new ideas have been provided to detect cyberbullying. Natural language processing is a theory-based computer technology that focuses on the automated analysis and expression of human language. This technology has been widely used in various fields, including sequence generation, machine translation, and recommender systems [3].

In recent years, there have been studies related to sentiment analysis and cyberbullying detection, namely research from [4] stated that nowadays, many people express their

DOI: 10.24507/icicelb.13.06.655

opinions with language that tends to be ambiguous and use complicated word choices. Often many words have a relationship with other words, even have similar meanings. Word embedding method can be used to find the similarity of words' meanings. Word embedding is a type of word representation that allows words that have similar meanings to be understood by machine learning algorithms [5]. There are several commonly used word embedding models, namely Word2Vec (Google), Glove (Stanford), and FastText (Facebook).

In previous research, [6] has tried to combine FastText word embedding with several classification algorithms such as Naïve Bayes, SVM, and XGBoost to detect spam in Indonesian-language Instagram posts. From the results of the research, FastText is quite helpful in increasing the accuracy of each classification algorithm. As a result, all classifiers combined with FastText have accuracy above 80%.

Based on the results of previous studies, this study will research cyberbullying detection in text conversations using the Word Embedding FastText method as the proposed model. In this study, the word embedding FastText model will be used because it managed to obtain the best accuracy compared to Word2Vec and Glove in the previous study [7]. In addition, the FastText model has advantages. One of them is the ability to handle words that we have never encountered before (Out of vocabulary words or known as OOV). For example, non-standard words such as "Optimization" will still get the vector. The Word2Vec library or the traditional one hot encoding technique described earlier will result in an error when receiving a word that is not in the dictionary [8].

The contribution of the paper can be summarized as the following points.

1) The extraction feature used is word embedding fast text. Word embedding is a useful method for representing words in vector form. This method can improve the performance of sentiment analysis; therefore, word embedding is widely used in research that discusses sentiment analysis.
2) The FastText method learns word representation by considering sub word information. Each word is represented as a set of n-gram characters. Thus, it can help capture the meaning of shorter words and allow embedding to understand the suffixes and prefixes of words in the dataset.
3) Using five algorithms and confusion matrix to see the best performance. The algorithms used for classification are SVM, Naïve Bayes, Random Forest, KNN, and Decision Tree.

The organization of the rest of this paper can be summarized as follows. Section 1 discusses about the research background of this study. Section 2 elaborates a lot regarding related works in cyberbullying detection. Section 3 describes the proposed solution by using FastText. Section 4 opens the results of the experiments as well as the related discussion. Lastly, Section 5 concludes all studies that were conducted in this research works.

2. **Related Works.** In this section, we discuss the related works in cyberbullying detection. By using word embedding, some representations of the semantic and syntactic relationship between words can be found. It allows us to capture the finer attributes and contextual cues inherent in human language [9]. After the publication of [10], word embedding became very popular [11], especially after the implementation of Word2Vec. Word2Vec obtains the representation of the word by looking at the context in which the word appears, thereby using the concept of distributed semantics. Two well-known slogans are included in the concept of distributed semantics. The first one comes from Firth (1968), which stated, "We can predict a word by the company it owns," and the second one comes from Harris (1954), which stated, "We understand the meaning of words through the context of the text".

An alternative toolkit for training embeddings is FastText [12]. The main difference between Word2Vec and FastText is that Word2Vec treats each word in the corpus as a separate subject and generates a vector for each word. FastText treats each word as consisting of character n-grams. Therefore, the vector of words consists of the sum of the n-grams of the character. The advantage of FastText is that it can accurately represent rare words because some of their n-grams are likely to appear in other words.

By using the Word2Vec tool, two different word embeddings architectures can be obtained. One is the continuous bag of words (CBOW). For the CBOW architecture, the input to the model is the preceding and following words of the central word, while the output of the model will be the central word. Therefore, we can think of the task as "predicting words based on context". The second architecture is skip-gram, where the input of the model is a word, and the output may be surrounding words. Therefore, the task here is to "predict the context of a given word". The skip-gram model is suitable for a small amount of training data and can even represent rare words. The training speed of CBOW is several times faster than skip-gram, and the accuracy for frequent words is slightly higher. An alternative toolkit to train embeddings is the FastText [12]. The main difference between Word2Vec and FastText is that Word2Vec treats each word in the corpus as a separate subject and generates a vector for each word. FastText treats each word as consisting of character n-grams. As a result, the vector of words consists of the sum of the n-grams of the character. The advantage of FastText is that it can represent rare words well because some of their n-grams may also appear in other words. Glove (global vectors for word representation) is an additional algorithm to train embeddings [13].

After training the word embeddings, it will be evaluated. The most common embeddings evaluation methods are i) the word semantic similarity method is based on the following fact: usually, the embeddings space can extract the word similarity by providing cosine similarity. [14] shows the similarity between this distance and human judgment; ii) the second more popular method is word analogy. It is based on the concept that arithmetic operations in a word vector space could be predicted by humans. For example, assuming words King, man, and woman, the predicted word should be Queen since the relation "King: man" is "property: sex"; thus, it must be found what is the property of female [15]; iii) another method for evaluation of embeddings is the concept categorization or word clustering. According to this method, the embeddings are clustering a set of given words according to pre-defined categories [16]; iv) an additional method to evaluate embeddings is synonym detection. For a given word, a set of words is provided, and the model must determine the synonym. This method is based on word semantic similarity; v) finally, embeddings can be evaluated with the detection of outliers in a group of words. This method is like the concept of categorization, where words are clustered in different groups.

[17] proposed a system to detect cyberbullying in Indonesian social media text using bidirectional encoder representations from transformers (BERT). This study uses real-world datasets and four pre-trained BERT models for comparison. The best model obtained is IndoBERT, with an average F1 score of 0.8229. The normalized data set produces better F1 scores for all models. Further fine-tuning the IndoBERT model using the normalized data set provides a higher F1 score of 0.84.

3. **Methodology.** To find out and analyze the pattern of actions taken by the perpetrator, it is necessary to identify cyberbullying in text conversations. From the results of the literature study that has been carried out, research on the analysis and detection of cyberbullying has been carried out to classify words that contain bullying. Figure 1 shows the stages carried out in this work.

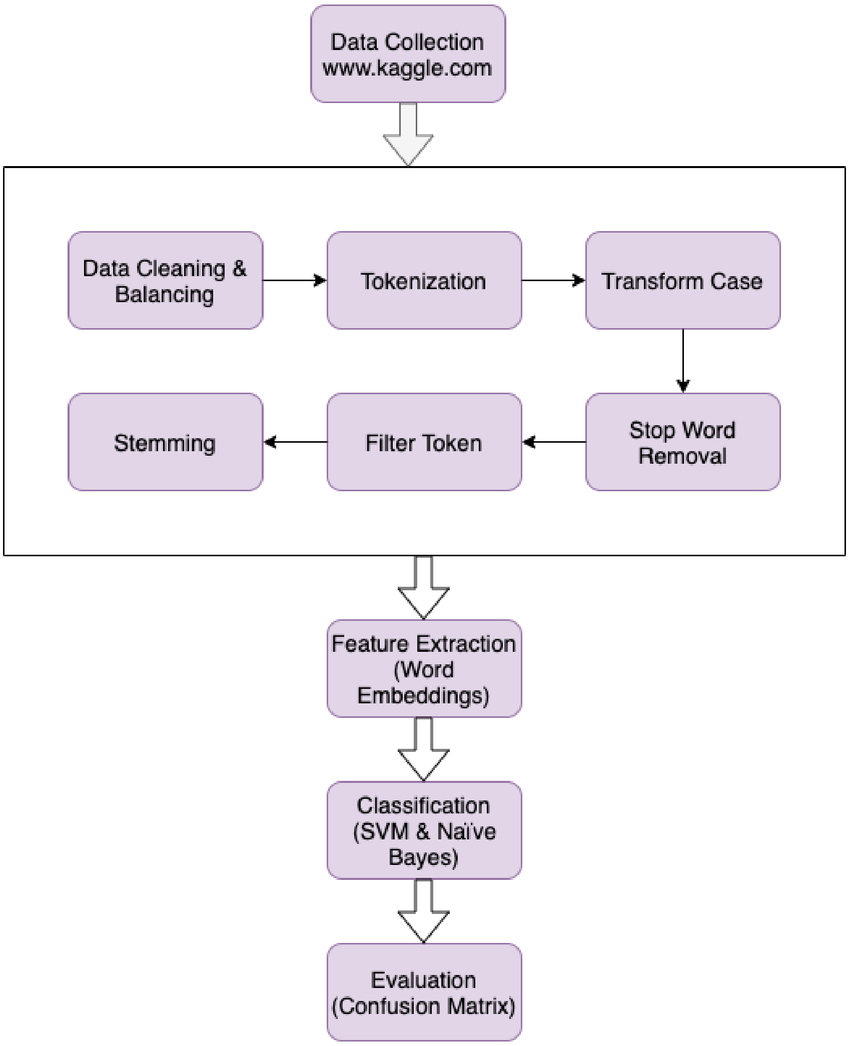
Data Collection
www.kaggle.com
Data Cleaning & Balancing
Tokenization
Transform Case
Stop Word Removal
Filter Token
Stemming
Feature Extraction (Word Embeddings)
Classification (SVM & Naïve Bayes)
Evaluation (Confusion Matrix)

FIGURE 1. Research method

3.1. **Data collection.** We collect a text conversation dataset from Kaggle (www.kaggle.com), which provides 1,600 conversations from Formspring.me. The fields Post is a combination of (Question and Answer), and Severity is used as a label in this research.

3.2. **Preprocessing.** In this phase, training data and test data are preprocessed to convert them into usable data. Each data will go through several stages in this preprocessing process, including

1) **Data cleaning & data balancing**
The total data retrieved from www.kaggle.com is 12,729, including 11,661 data with non-cyberbullying label and 1,068 data labeled cyberbully. Data cleaning is done with Python by eliminating conversations that have total characters under 15 letters to delete meaningless words like "haha", "hehe", "wkwk", "emm", "umm". Also, because of the heavy imbalance between the two classes (cyberbully and non-cyberbully), the amount of data used is adjusted to 1,600 data in order to balance the dataset (800 labeled cyberbully and 800 labeled non-cyberbully).
2) **Tokenization**
Tokenization is the process of dividing text or conversations, which can be in the form of a sentence, paragraph, or document, into tokens or certain parts.
3) **Transform case**
Convert to lower case for easier processing. The purpose is to not distinguish between uppercase and lowercase letters.

4) **Stop word removal**
Using the stop word filter (English), unnecessary words in each text conversation are deleted according to the English vocabulary.

5) **Filter token**
The number of characters selected by the token filter is between 3 and 25 because words with less than 3 characters are stop words, and characters with more than 25 characters are rarely used.

6) **Stemmer**
Convert words in text conversations into basic words.

3.3. **Feature extraction.** Our proposed model to detect cyberbullying uses FastText word embeddings. It proved to be effective in extracting similar features with similar word representations with the same meaning. Based on experiments and several previous studies [7], which compared the performance of Word2Vec, Glove, and FastText, the best performance from the experiment was obtained by using the FastText word embedding model.

We propose FastText and similarity word detection methods to analyze cyberbullying in conversation in fromspring.me. FastText is a library released by Facebook that can be used for word embeddings. FastText itself is a development of the Word2Vec library, which has long been known as a library for word embeddings. There are several advantages of FastText over Word2Vec. One of them is FastText's ability to handle words that we have never encountered before (Out of vocabulary word, also known as OOV) [7].

$$f_{subword} : (v(c_1, \ldots, c_n)) \rightarrow h \tag{1}$$

FastText syllable made from a vocabulary and sequence character $(c_1, \ldots, c_n)$ into vector h. Sequence language character indicate composition of information from word meaning [18].

Figure 2 shows the CBOW model with multi-word context settings. When calculating the output of the hidden layer, the CBOW model does not directly copy the input vector of the input context word but takes the average value of the vector of the input context

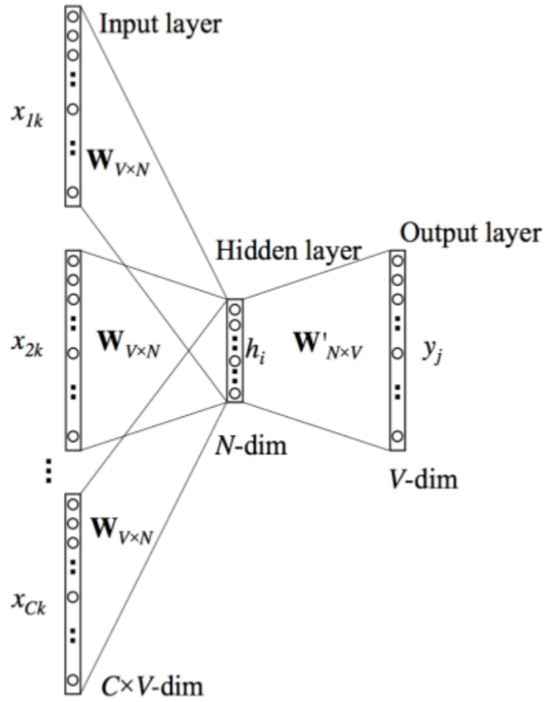

FIGURE 2. Continuous bag-of-words model

word and uses the product of the input → hidden weight matrix and the average vector as the output [19]. CBOW's training speed is faster than skip-gram, but skip-gram is more accurate than CBOW [20]. This model uses context to predict the target word. CBOW has a shorter training time and has slightly better accuracy for frequent words.

3.4. **Classification.** There are many types of machine learning algorithms used for classification in sentiment analysis. In this stage, there are five scenarios used to see the best performance. The classification methods used are SVM, Naïve Bayes, Random Forest, KNN, and Decision Tree. Therefore, they are the most useful and reliable analysis algorithms. The researcher chose these five algorithms because they generally performed well on several cases or datasets.

3.5. **Evaluation.** For the evaluation and validation, the cross-validation method is used to measure the performance of 2 classes using the confusion matrix. The dataset is randomly divided into 80% training data and 20% testing data. Experiments are carried out by classifying them into two classes, namely:

1) Class Yes: Contains conversations that involve cyberbullying
2) Class No: Contains conversations that are not cyberbullying

4. **Experiment Result and Discussion.** In this section, we discuss the results of testing five scenarios and the classification performance using the SVM, Naïve Bayes, Random Forest, Decision Tree, and KNN methods. The following are the results of five classification performances evaluated from three aspects: recall, precision, and F1-Score.

Table 1. Result of testing five scenarios on recall, precision and F1-Score

No	Classification	Recall	Precision	F1-Score
1	SVM	0.91	0.93	0.93
2	Naïve Bayes	0.68	0.69	0.69
3	Random Forest	0.88	0.88	0.88
4	Decision Tree	0.78	0.80	0.77
5	KNN	0.73	0.73	0.72

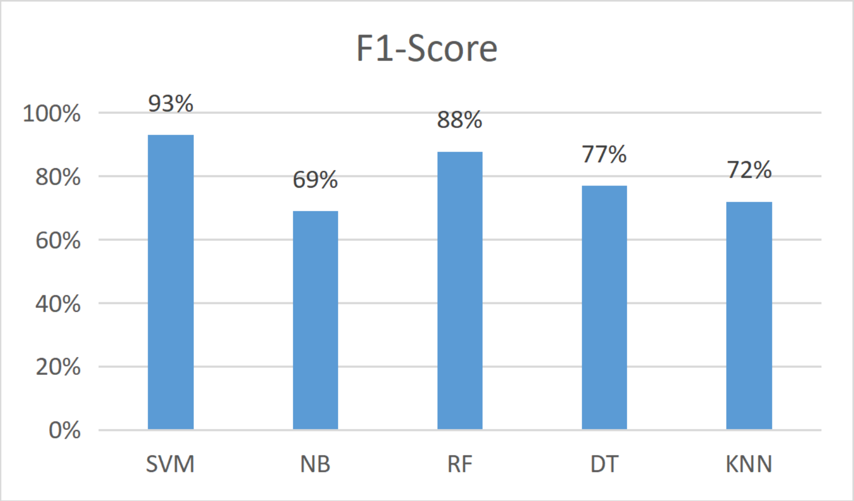

Figure 3. Result of performance classification

From Table 1 and Figure 3, it can be explained that the best performance of the five scenarios in classifying text conversations is the SVM method with the highest F1-Score of 0.93, compared to the performance of the Random Forest, Decision Tree, KNN, and Naïve Bayes.

The following Table 2 describes the results of the SVM algorithm in the form of a confusion matrix:

Table 2. Result of confusion matrix 2 class

SVM	Pred No	Pred Yes
Actual. No	160	0
Actual. Yes	28	132

1) Among 160 testing data labeled as "Yes", there are 132 data correctly predicted as "Yes", indicated as cyberbullying. While 28 data contained prediction errors.
2) For the 160 test data that are labeled as "No", there are also 160 data that match the prediction "No", indicated as non-cyberbullying. While 0 data contained prediction errors.

Based on Table 2, the SVM algorithm works in accordance with the purpose of our research, which is to detect sentences that are classified as cyberbullying and non-cyberbullying automatically.

5. Conclusions.

Several methods can be used to represent text in vectors, one of which uses word embeddings. In this study, observations were made on cyberbullying and word embeddings of FastText to detect cyberbullying. FastText is built on the Word2Vec model, which is based on sub words or syllables. From the results of the research that has been carried out in this paper, it can be taken as follows.

1) This study succeeded in building a model to detect cyberbullying using FastText word insertion.
2) The best classification performance is the SVM method with the highest F1-Score 0.93 compared to the Random Forest, Decision Tree, KNN, and Naïve Bayes methods.

It is recommended for further research to detect cyberbullying using a data collection of Indonesian conversations with the FastText word embedding method because the classification of cyberbullying in Indonesian is more interesting and challenging.

REFERENCES

[1] R. M. Kowalski, G. W. Giumetti, A. N. Schroeder and M. R. Lattanner, Bullying in the digital age: A critical review and meta-analysis of cyberbullying research among youth, *Psychological Bulletin*, vol.140, no.4, pp.1073-1137, 2014.

[2] S. Cook, *Comparitech*, https://www.comparitech.com/internet-providers/cyberbullying-statistics/#Cyberbullying_facts_and_statistics_for_2018-2020, Accessed on January 31, 2021.

[3] H. Rosa, N. Pereira, R. Ribeiro, P. C. Ferreira, J. P. Carvalho, S. Oliveira, L. Coheur, P. Paulino, A. M. Veiga Simão and I. Trancoso, Automatic cyberbullying detection: A systematic review, *Computers in Human Behavior*, vol.93, pp.333-345, 2019.

[4] P. Vora, M. Khara and K. Kelkar, Classification of tweets based on emotions using word embedding and random forest classifiers, *International Journal of Computer Applications*, vol.178, no.3, pp.1-7, 2017.

[5] K. Benoit, K. Munger and A. Spirling, Measuring and explaining political sophistication through textual complexity, *American Journal of Political Science*, vol.63, no.2, pp.491-508, 2019.

[6] M. H. Krishna, K. Rahamathulla and A. Akbar, A feature based approach for sentiment analysis using SVM and coreference resolution, *International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, pp.397-399, 2017.

[7] A. Nurdin, B. A. S. Aji, A. Bustamin and Z. Abidin, Performance comparison of word embedding Word2Vec, Glove, and FastText in text classification, *Journal of Compact Techno*, vol.14, no.2, pp.74-79, 2020.

[8] K. Wang, Y. Cui, J. Hu, Y. Zhang, W. Zhao and L. Feng, Cyberbullying detection, based on the FastText and word similarity schemes, *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol.20, no.1, pp.1-15, 2020.

[9] D. Chatzakou, N. Kourtellis, J. Blackburn, E. D. Cristofaro, G. Stringhini and A. Vakali, Mean birds: Detecting aggression and bullying on Twitter, *Proc. of the 2017 ACM on Web Science Conference*, pp.13-22, 2017.

[10] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, *arXiv.org*, arXiv: 1301.3781, 2013.

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, *Proc. of the 26th International Conference on Neural Information Processing Systems*, vol.2, pp.3111-3119, 2013.

[12] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, Bag of tricks for efficient text classification, *Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, pp.427-431, 2017.

[13] J. Pennington, R. Socher and C. D. Manning, Glove: Global vectors for word representation, *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.1532-1543, 2014.

[14] M. Faruqui and C. Dyer, Community evaluation and exchange of word vectors at wordvectors.org, *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp.19-24, 2014.

[15] M. Baroni, G. Dinu and G. Kruszewski, Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp.238-247, 2014.

[16] Y. Li, R. Zheng, T. Tian, Z. Hu, R. Iyer and K. Sycara, Joint embedding of hierarchical categories and entities for concept categorization and dataless classification, *Proc. of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, pp.2678-2688, 2016.

[17] A. Candra, Wella and A. Wicaksana, Bidirectional encoder representations from transformers for cyberbullying text detection in Indonesian social media, *International Journal of Innovative Computing, Information and Control*, vol.17, no.5, pp.1599-1615, 2021.

[18] J. Hewitt, *Finding Syntax with Structural Probes*, https://nlp.stanford.edu//~johnhew//structural-probe.html?utm_source=quora&utm_medium=referral#the-structural-probe, Accessed on August 18, 2021.

[19] X. Rong, *Word2Vec Parameter Learning Explained*, https://www.researchgate.net/publication/268226652_word2vec_Parameter_Learning_Explained, Accessed on August 12, 2021.

[20] C.-Y. Chang, S.-J. Lee and C.-C. Lai, Weighted Word2Vec based on the distance of words, *International Conference on Machine Learning and Cybernetics (ICMLC)*, Ningbo, China, vol.2, pp.563-568, 2017.