Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
15/1 (2022), 39-46. DOI: http://dx.doi.org/10.21609/jiki.v15i1.1044

# Sentiment Analysis of COVID-19 Vaccines in Indonesia on Twitter Using Pre-Trained and Self-Training Word Embeddings

Kartikasari Kusuma Agustiningsih[1], Ema Utami[2], and Omar Muhammad Altoumi Alsyaibani[3]

[1,2,3]Magister of Informatics Engineering, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
*E-mail: kartikasarikusuma@students.amikom.ac.id[1], ema.u@amikom.ac.id[2], omar@smkn2banjarbaru.sch.id [3]*

**Abstract**

Sentiment analysis regarding the COVID-19 vaccine can be obtained from social media because users usually express their opinions through social media. One of the social media that is most often used by Indonesian people to express their opinion is Twitter. The method used in this research is Bidirectional LSTM which will be combined with word embedding. In this study, fastText and GloVe were tested as word embedding. We created 8 test scenarios to inspect performance of the word embeddings, using both pre-trained and self-trained word embedding vectors. Dataset gathered from Twitter was prepared as stemmed dataset and unstemmed dataset. The highest accuracy from GloVe scenario group was generated by model which used self-trained GloVe and trained on unstemmed dataset. The accuracy reached 92.5%. On the other hand, the highest accuracy from fastText scenario group generated by model which used self-trained fastText and trained on stemmed dataset. The accuracy reached 92.3%. In other scenarios that used pre-trained embedding vector, the accuracy was quite lower than scenarios that used self-trained embedding vector, because the pre-trained embedding data was trained using the Wikipedia corpus which contains standard and well-structured language while the dataset used in this study came from Twitter which contains non-standard sentences. Even though the dataset was processed using stemming and slang words dictionary, the pre-trained embedding still can not recognize several words from our dataset.

**Keywords:** Sentiment Analysis, Twitter, Bidirectional LSTM, Word Embedding, fastText, GloVe

## 1. Introduction

The COVID-19 virus (Corona Virus Disease 2019) began to spread throughout the world at the end of 2019. This virus originated in a city in China that attacks the human respiratory system. Based on data from the Worldometer in December 2022, there were more than 271 million cases of transmission with a death rate of more than 5 million. The vaccination process was initiated by the United States and Israel since December 2020 and was followed by other countries, including Indonesia [1]. The results of the study [2] revealed that there are several countries where public acceptance of vaccines is low. Sentiment analysis can be used to understand people's perceptions on social media by analyzing their opinions on various topics [3]. Therefore, a classification technique was developed to classify a developing opinion, both on social media and other mass media.

An important factor in the field of Natural Language Processing (NLP), especially sentiment analysis is the data vectorization process. There are several techniques that can be used to vectorize data such as Term Frequency–Inverse Document Frequency (TF-IDF) and Word Embedding. In many studies related to sentiment analysis with the Indonesian language corpus, the TF-IDF method is the most widely used because TF-IDF is a data vectorization technique that is quite simple and easy to apply in a study such as that conducted by authors [4], [5] and [6].

On the other hand, research on sentiment analysis has begun to emerge using the word embedding technique. Research conducted by [7] compared the performance of Word2Vec, Global Vectors for Word Representation (GloVe) and fastText embedding which were tested on the Convolutional Neural Network (CNN) method. The models were tested using 20 newsgroups dataset and the English-language Reuters Newswire dataset. The results of their research show that the CNN model which used fastText word embedding has better performance than CNN

model which used other two word-embeddings. The results of the study [8] also proved the same results where fastText word embedding used in 4 different methods which were Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Gated Recurrent Unit (GRU) and Bidirectional GRU (Bi-GRU). In the study, models which used fastText embedding produced better performance than the same models which used GloVe embedding.

Another study implemented word embedding is a study conducted by [9] where fastText, GloVe and Word2vec embedding were used with LSTM to detect emotions from Twitter users with Indonesian language data. fastText and Word2vec embedding were implemented using the Gensim library, while the GloVe used were pre-trained GloVe. From the results of the study, fastText and Word2vec word embedding obtained similar accuracy results of 73.15%. Furthermore, this study recommended to use Bi-LSTM in the future work.

Differently, research [10] used Word2Vec embedding to vectorize dataset for classifying public sentiment on Fast Food Companies from Twitter. The dataset was in English. CNN, Bi-LSTM and CNN combined with Bi-LSTM. The result of this study shows that Bi-LSTM model provided the highest accuracy compared to other models.

Study [7] and [10] studied word embedding in English dataset. Study [8] also used English dataset which was translated to Bahasa. Only study [9] used Bahasa dataset. Furthermore, unlike study [9] which worked on non-formal language data, both studies [7] and [8] used formal language dataset. On the other hand, study [9] compared performance of fastText, GloVe and Word2Vec embeddings. The study mentioned that GloVe vector used was pre-trained vector. Yet, it did not explain whether fastText and Word2Vec used was also pre-trained or not. It is important to understand the embedding vectors used to translate document into numbers because pre-trained embedding vectors usually were trained on formal language corpus while sentiment analysis study works on non-formal language data. It is possible that several words in dataset are not vectorized correctly by pre-trained embedding vector because it was trained on formal language corpus and did not contain the non-formal words from Twitter. Thus, this research aims to study this problem.

Study [9] stemmed the dataset to return all words into original form. The dataset used was a non-formal language data. Yet, the authors did not discuss whether the stemming process improved the performance of the model or not. Therefore, this study also aims to test the effect of the stemming process both in pre-trained vector and in self-trained vector.

The fastText and GloVe embedding used in this study were trained on the dataset obtained from Twitter to generate self-trained embedding vectors. The pre-train fastText word vector was published by the author [11] and the pre-train GloVe embedding was published by the author [12] were also used as comparisons. Both pre-train vectors used were trained using the Indonesian Wikipedia Corpus. The result of study [10] and recommendation of authors [9] suggest Bi-LSTM architecture to classify public sentiment on Twitter data. Thus, this study used it as classification method.

## 2. Methodology

### 2.1 Data Collection

The data collection used a scraping technique with the SNScrape tool created by author [13]. The tool can retrieve data from Twitter which more than 7 previous days. The SNScrape tool requires the user to have a Twitter Developer account. The data was taken using the keyword "covid vaccine" and scaped specifically on September 2021.

The data labelling process was done manually. The data collection had to be conducted several times due to high volume of data. SNScrape will be blocked by Twitter when scaping very large number of tweets. This caused tweet duplication on data. This problem was address on data preprocessing stage.

### 2.2 Data Preprocessing

The first step in data preprocessing is data cleaning. All tweet data that have the same tweet id were deleted. The existence of duplicate tweets occurred because data collection must be done several times. The next step is the process of identifying news accounts and organizations. All tweets from these accounts were deleted because they did not represent the sentiments of the public and their objectivity toward vaccine was unknown.

Data preprocessing techniques can be done to convert data into structured data considering that data from Twitter is usually unstructured data which will be difficult to classify by algorithms. The most frequently used preprocessing technique in sentiment analysis research on COVID-19 vaccines is stopwords removal and remove punctuation [14]. The next cleaning process is the removal of punctuation, whitespaces, hashtags, “RT” characters, hyperlinks and mentions.

In the next stage, case folding was implemented on dataset. Case folding process changed all characters to lowercase. It was needed to be done because sometimes Twitter uses write in incorrect case. If a word is written in several different ways

in a corpus, the word will produce a different vector when vectorized.

Some Twitter users in Indonesia use non-standard language (slang) in making tweets [15]. This will make it difficult for the model to understand the language in the tweet. To overcome this, every slang word must be returned to the original word form using the Alay Dictionary published by the author [16]. Fig.1 shows the preprocessing flow

Indonesian has many phrases of two or more words that refer to a single thing such as the words "rumah sakit", "turun tangan" or "adu domba". A model must be able to distinguish between the word "rumah" and "sakit" with the phrase "rumah sakit". In the tokenization process, space were used as separators between words [17]. In this study, the dividing space in the phrases was omitted. To obtain all phrases in Indonesian, data were collected from [18] and [19]. From the two data sources, all phrases were taken and then combined. This data was used as a benchmark in the process of removing the dividing space between words in phrases.

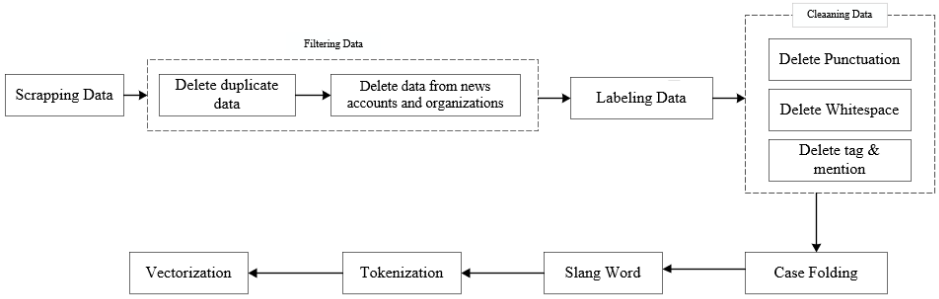


**Fig 1**. Data Prepocessing Flow

## 2.3 Vectorization

In this study, two types of datasets were prepared, datasets that was stemmed and datasets that do not. Stemming process changes an affixed word into a root word [20]. The stemming process was conducted based on the work of [21] because currently, it is the most used stemming algorithm for Bahasa. In this study, the effect of stemming on model performance was also tested in both word embeddings.

In order for a model to process a sentence in a natural language processing study, a word must be converted into a vector form. GloVe and fastText are techniques that can convert words into vectors. GloVe is an algorithm introduced by [22]. It is an improvement from the matrix factorization–based representations of words and the Skip-gram model where the matrix factorization–based representations of words method is not very good at representing words with respect to their analogous properties [23]. GloVe is based on the factorization technique [24]. It is referred to as a modification of Word2vec [25].

Skip-Gram and CBoW train in different way than the matrix-based factorization method. For example, in the LDA technique is used to create topic modeling, the previous text must be processed first by encoding each word with statistical information that represents the word in the context of the entire text. Thus, the one-hot encode vector cannot understand the same kind of complexity when using the Skip-Gram and CBoW methods. GloVe is very efficient in capturing the semantic details of a word in its vector representation, but very inefficient in carrying out sentiment analysis [24]. Several solutions have been proposed by researchers by modifying the C&W model [26].

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2 \quad (1)$$

The GloVe method is formulated by the authors [22] into Equation 1. $V$ represents corpus size, $b$ represents bias, w represents weight, and $X$ represents word processed on matrix i x j. GloVe architecture was depicted by authors [27] as in Fig 2.

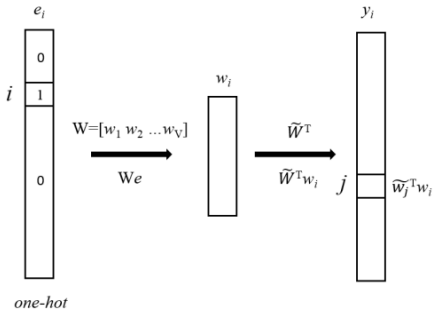


**Fig. 2**. GloVe architecture [27]

fastText is a Facebook-owned library used to generate efficient word representations and provide support for text classification [28]. fastText is an update model of a pre-trained Skip-gram [25]. The

same thing was also stated by [29]. fastText is generally used to solve sentence classification and word representation problems to be more efficient and faster than the Word2vec and GloVe methods [29]. fastText uses Skip-gram based approach where each word is represented as a bag of character n-grams [30]. fastText looks at the provided corpus of text and forms a high-dimensional vector space model, in which it tries to summarize as much meaning as possible. The purpose of creating a vector space is for vectors of similar words to be close together [28]. In fastText, this word vector is then stored in two files, *.bin file and *.vec file [31]. fastText function is represented in Equation 2. $S$ represents scoring function, $w$ represents weight, $l$ represents log(1 + e − x ), and $n$ represents number of words in corpus.

$$\sum_{t=1}^{T}\left[\sum_{c\in\mathcal{C}_t}\ell(s(w_t,\ w_c))+\sum_{n\in\mathcal{N}_{t,c}}\ell(-s(w_t,\ n))\right] \quad (2)$$

Both GloVe and fastText embedding can be trained using custom corpus. Wikipedia provides a very large corpus for various languages. One of them is Indonesian. In this study, the pre-train word embedding vectors which were trained on the Wikipedia corpus and the embedding vectors which were trained on Twitter dataset obtained were tested. The GloVe pre-train vector for Bahasa was published by the author [12] while the pre-train fastText vector for Bahasa was published by the author [11].

### 2.4 Classification

To test the performance of Word Embeddings, the Bidirectional LSTM network was created for every test scenario. The dataset obtained was divided into 2 parts, training-validation and test data with a distribution of 80:20. The models was created using the TensorFlow and Keras frameworks. The performance of the models during training were monitored using Early Stopping with a patience value 3. The Early Stopping monitored validation loss during training process. If the Early Stopping found that validation loss increased for 3 iterations during training, then the training model would be stopped.

All embeddings implemented, both pre-train and self-trained, used 300 dimensions, except for the pre-train GloVe vector created by the author [12] which has 50 dimensions. In all scenario, the same Bidirectional LSTM architecture was used. First of all, the data will enter into the Input layer and then forwarded to the Embedding layer. In the next stage, 3 Bidirectional LSTM layers were implemented with 128, 64 and 32 number of neurons, respectively. In the last layer, Dense layer was deployed contained 3 neurons with Softmax activation function because the dataset had 3 classes. To avoid the model from being overfitted, the Dropout layer was implemented in between each layer. The weight of each neuron was optimized during the training process using Adam developed by the authors [32]. According to the author [33], Adam's optimization function can make the model achieve the optimal weight value in a short number of training iterations. During the training process, 30% of the training data was used as validation data. The batch size value used was 64.

## 3. Result and Discussion

### 3.1 Dataset

The data collected from Twitter amounted to 11414 tweets. After the cleaning stage, the remaining data amounted to 6547 tweets with a composition of 4476 neutral tweets, 1742 positive tweets and 329 negative tweets. The composition of the dataset is depicted in Fig 3. Based on Figure 3, it can be seen that in September 2021, the level of public acceptance of the COVID-19 vaccine was quite good.

The high number of tweets with neutral sentiments by the public is indeed a discussion about the COVID-19 vaccine, yet it has not led to positive or negative sentiments. These discussions were more about sharing information about the COVID-19 vaccination process.

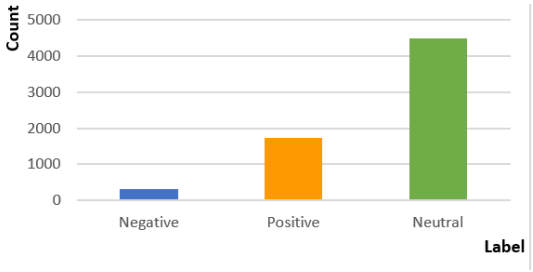


**Fig. 3**. Dataset Composition

It is also can be inferred from Fig. 3 that the label on the data is imbalanced. Therefore, prior to the classification, oversampling was carried out on the data so that the functions in the model that had been trained would not be skewed [34] towards neutral and positive only.

After the oversampling process was done, the data was stored into two types. The first type is the data that was processed using stemming and the second data was the data that was not. The stemming algorithm used was the one published by the authors [21]. From these two data, GloVe and fastText self-trained vector were created with the same training parameter for both embeddings. The

embedding size used was 300 and window size was 5. They were trained in 100 iterations. Vectors generated from the training were used on classification stage.

### 3.2 Classification Result

The initial classification was conducted using self-trained embedding which consisted of 4 test scenarios. The results of the initial classification test are shown in Fig. 3. X-Axis represents the test scenario and the use of stemming. Y-Axis represents the accuracy of each test scenario.

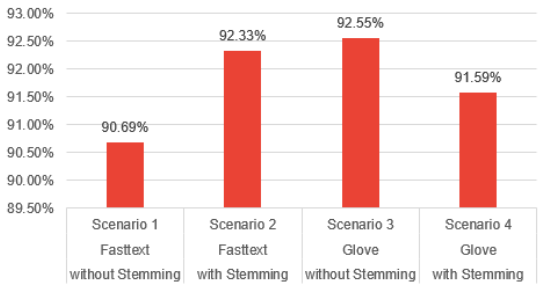


**Fig. 4.** Test Accuracy of Scenario 1 to 4

It can be seen from Fig. 4 that the fastText word embedding has better accuracy when using stemmed dataset. On the other hand, GloVe word embedding has better accuracy when using unstemmed dataset. GloVe word embedding accuracy is slightly higher than fastText word embedding with an accuracy difference of 0.2234%. This result is in contrast to the results of the study [7] and [8] where fastText word embedding produced higher accuracy than GloVe word embedding. It is need to be noted that datasets used by authors [7] was in English and dataset used by author [8] was an English fake news data which translated into Bahasa. Unlike data gathered from Twitter, all datasets that they used had good language structure.

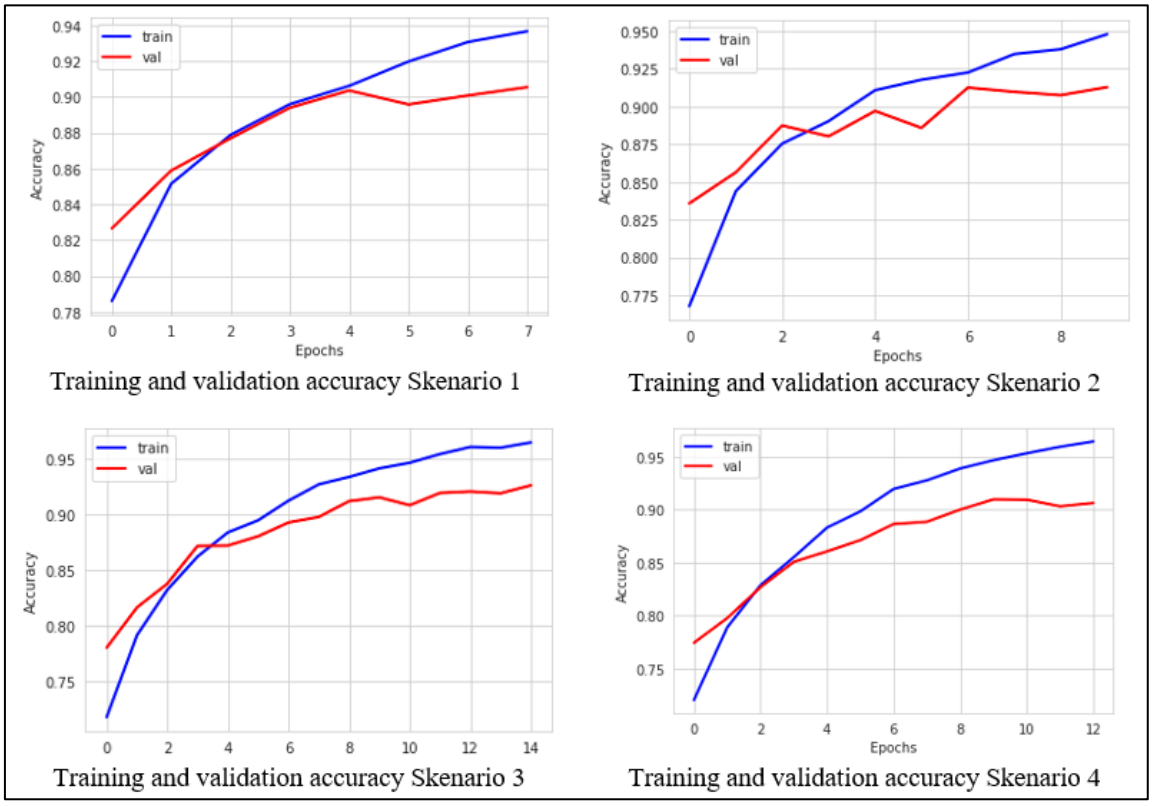


**Fig. 5**. Training history of Scenario 1 to 4

Fig. 5 illustrates the training and validation accuracy of models during training. The blue line represents training accuracy and validation accuracy is represented by the orange line. It can be identified from Fig. 5 that the validation accuracy in all scenarios started out higher than the

training accuracy, but in the 4th iteration the increase in validation accuracy slowed down. Validation accuracy stops increasing at different iterations in each scenario. This training was stopped automatically by Early Stopping which monitored the validation loss. If in the last 3 iterations there is no decrease in the validation loss (patience = 3), then the training will be stopped.

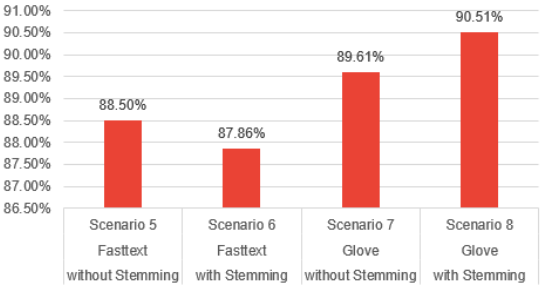


**Fig.6**. Test Accuracy of Scenario 5 to 8

The experiment was continued by testing the pre-train fastText and GloVe word embeddings. The results of this test can be seen in Fig. 6. The X-Axis represents the test scenario and the use of stemming while the Y-Axis represents the accuracy value of each test scenario. From Fig 6, it can be seen that the results of this test are the opposite of the previous test. In the results of this test, it is known that fastText produced higher accuracy on unstemmed dataset while GloVe produced higher accuracy on stemmed dataset. The highest accuracy was generated by scenario 8 where the word embedding used was GloVe on the stemmed dataset. However, in general, the accuracy produced in scenarios using pre-train word embedding is lower than the self-trained word embedding.

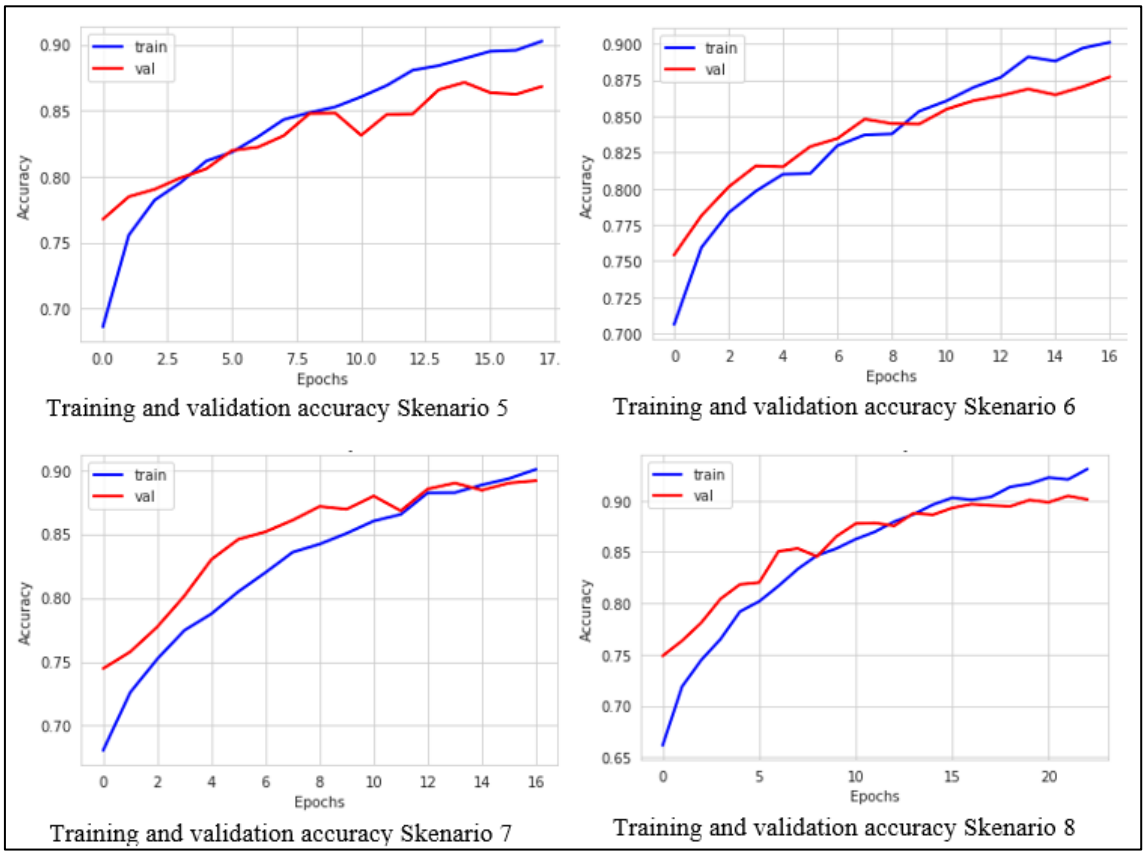


**Fig. 7** Training history of Scenario 5 to 8

Fig.7 illustrates the increase in training accuracy and model validation in scenarios 5, 6, 7 and 8. Training accuracy is represented by a blue line and validation accuracy is represented by an orange line. From Figure 6, it can be seen that the model that used pre-train word embedding performed longer training to achieve optimal accuracy when compared to the model that used self-trained word embedding. Testing of scenario 8 spent the most iterations, which was 23 epochs. Although these four training scenarios were trained in a larger number of iterations, the accuracy of the

model was not better than the model that used self-trained word embedding. This fact is in line with the results of research [35] where self-training embedding has a better performance than pre-trained embedding.

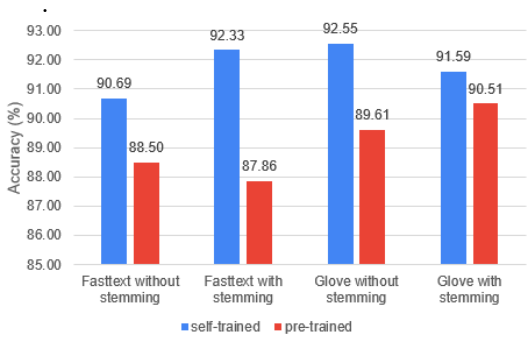


**Fig.8**. Comparison Between self-trained and pre-trained embedding

The comparison between self-trained and pre-trained embedding shown in Fig.8. The pre-trained word embedding used in this study were pre-trained word embeddings that were trained using the Wikipedia corpus which contains formal and well-structured language. Although the datasets was processed using algorithm published by [21] and Alay Dictionary published by [16], it is still possible that several words were not handled properly. Study [36] found that stemming algorithm published by [21] cannot handle slang words properly. Thus they could not be recognized by the pre-trained word embedding made by author [12] and authors [11]. Words that are not recognized by word embedding will be assigned the same vector value [37]. Thus, it is possible for the model not to recognize words that may play an important role in determining the sentiment of a tweet. Word embedding trained with datasets originating from the same data source as those to be classified will be able to properly identify each word, both formal and slang words, so that each word will produce a different vector value.

## 4. Conclusion and Future Work

In this study, we tested the use of word embedding on sentiment classification using the Bidirectional LSTM model. Word embeddings tested were fastText and GloVe. The effect of stemming usage toward performance of the model was also inspected. Sentiment data was collected from Twitter in September 2021 with the keyword "covid vaccine" to see publis's sentiments about vaccines in that month. Most people have neutral sentiments, as many as 4476 tweets. The positive sentiment were 1742 tweets and negative sentiments were 329 tweets.

From the results of the classification test, it is known that self-trained word embedding can make the model produce higher accuracy than using pre-train word embedding. Models that used GloVe word embedding resulted in higher accuracy on unstemmed datasets. Different results were produced by the model using fastText word embedding where higher accuracy was obtained when using stemmed dataset. The difference in accuracy between fastText and GloVe word embedding was very small where GloVe word embedding produced slightly higher accuracy than FastText.

This study contributes to confirm that word embedding that is trained on data with the same characteristic with test data can perform better performance compared to pre-trained word embedding which is trained on general corpus data. Therefore, we recommend the sentiment analysis study on non-formal language dataset such as emotion detection, sentiment classification or sarcasm detection which used word embedding to generate self-trained word embedding vector.

In future study, we aim to test contextual word representation such as BERT, both using self-trained and pre-train vectors. In addition, research will also be conducted on datasets containing more comprehensive tweets.

## References

[1] H. Ritchie *et al.*, "Coronavirus (COVID-19) Vaccinations," 2021. https://ourworldindata.org/covid-vaccinations.

[2] M. Sallam, "Covid-19 vaccine hesitancy worldwide: A concise systematic review of vaccine acceptance rates," *Vaccines*, vol. 9, no. 2. 2021, doi: 10.3390/vaccines9020160.

[3] D. A. Nurdeni, I. Budi, and A. B. Santoso, "Sentiment Analysis on Covid19 Vaccines in Indonesia: From the Perspective of Sinovac and Pfizer," *3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021*, pp. 122–127, 2021, doi: 10.1109/EIConCIT50028.2021.9431852.

[4] Pristiyono, M. Ritonga, M. A. Al Ihsan, A. Anjar, and F. H. Rambe, "Sentiment analysis of COVID-19 vaccine in Indonesia using Naïve Bayes Algorithm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1088, no. 1, 2021, doi: 10.1088/1757-899x/1088/1/012045.

[5] W. Yulita, E. D. Nugroho, and M. H. Algifari, "Analisis Sentimen Terhadap Opini Masyarakat Tentang Vaksin Covid - 19 Menggunakan Algoritma Naïve Bayes Classifier," vol. 2, no. 2, pp. 1–9, 2021.

[6] B. Laurensz and Eko Sediyono, "Analisis Sentimen Masyarakat terhadap Tindakan Vaksinasi dalam Upaya Mengatasi Pandemi Covid-19," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 10, no. 2, pp. 118–123, 2021, doi: 10.22146/jnteti.v10i2.1421.

[7] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," *J. Tekno Kompak*, vol. 14, no. 2, p. 74, 2020, doi: 10.33365/jtk.v14i2.732.

[8] R. Adipradana, B. P. Nayoga, R. Suryadi, and D. Suhartono, "Hoax analyzer for indonesian news using rnns with fasttext and glove embeddings," *Bull. Electr. Eng. Informatics*, vol. 10, no. 4, 2021, doi:

10.11591/eei.v10i4.2956.

[9] M. A. Riza and N. Charibaldi, "Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text," *Int. J. Artif. Intell. Robot.*, vol. 3, no. 1, pp. 15–26, 2021, doi: 10.25139/ijair.v3i1.3827.

[10] G. Abdalla and F. Özyurt, "Sentiment Analysis of Fast Food Companies with Deep Learning Models," *Comput. J.*, vol. 64, no. 3, pp. 383–390, 2021, doi: 10.1093/comjnl/bxaa131.

[11] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors for 157 Languages," 2018.

[12] I. Hanif, "Klasifikasi Perintah Bahasa Natural Menggunakan Global Vectors for Word Representations (GloVe), Convolutional Neural Networks, dan Teknik Transfer Learning pada Aplikasi Chatbots," Institut Teknologi Sepuluh Nopember, 2018.

[13] JustAnotherArchivist, "snscrape," 2021. https://github.com/JustAnotherArchivist/snscrape.

[14] K. K. Agustiningsih, E. Utami, and H. Al Fatta, "Sentiment Analysis of COVID-19 Vaccine on Twitter Social Media: Systematic Literature Review," in *2021 IEEE 5th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2021, pp. 121–126, doi: 10.1109/ICITISEE53823.2021.9655960.

[15] E. Utami, I. Oyong, S. Raharjo, A. Dwi Hartanto, and S. Adi, "Supervised learning and resampling techniques on DISC personality classification using Twitter information in Bahasa Indonesia," *Appl. Comput. Informatics*, vol. ahead-of-p, no. ahead-of-print, Jan. 2021, doi: 10.1108/ACI-03-2021-0054.

[16] N. Aliyah Salsabila, Y. Ardhito Winatmoko, A. Akbar Septiandri, and A. Jamal, "Colloquial Indonesian Lexicon," in *2018 International Conference on Asian Language Processing (IALP)*, 2018, pp. 226–229, doi: 10.1109/IALP.2018.8629151.

[17] E. Utami, A. D. Hartanto, S. Adi, R. B. Setya Putra, and S. Raharjo, "Formal and Non-Formal Indonesian Word Usage Frequency in Twitter Profile Using Non-Formal Affix Rule," in *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*, 2019, vol. 1, pp. 173–176, doi: 10.1109/ICORIS.2019.8874908.

[18] Wikipedia, "Daftar frasa idiomatis dalam bahasa Indonesia," 2021. .

[19] S. R. Umami, "Kamus Besar Bahasa Indonesia Edisi IV," 2021. .

[20] J. B. Lovins, "Development of a stemming algorithm.," *Mech. Transl. Comput. Linguist.*, vol. 11, no. 1–2, pp. 22–31, 1968.

[21] M. Nazief, B. A. A. & Adriani, "Confix- stripping: Approach to Stemming Algorithm for Bahasa Indonesia," *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 38, no. 4, 2005.

[22] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[23] T. Beysolow II, *Applied natural language processing with python*. Springer, 2018.

[24] S. Poria, A. Hussain, and E. Cambria, *Multimodal Sentiment Analysis*. Springer Nature, 2018.

[25] T. T. Mengistie and D. Kumar, "Deep Learning Based Sentiment Analysis On COVID-19 Public Reviews," in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2021, pp. 444–449.

[26] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, 2011.

[27] C. Liu, P. Zhang, T. Li, and Y. Yan, "Semantic Features Based N-Best Rescoring Methods for Automatic Speech Recognition," *Appl. Sci.*, vol. 9, no. 23, p. 5053, 2019.

[28] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017.

[29] A. G. D'Sa, I. Illina, and D. Fohr, "BERT and fastText Embeddings for Automatic Detection of Toxic Speech," *Proc. 2020 Int. Multi-Conference Organ. Knowl. Adv. Technol. OCTA 2020*, 2020, doi: 10.1109/OCTA49274.2020.9151853.

[30] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv Prepr. arXiv1612.03651*, 2016.

[31] J. Bhattacharjee, *FastText Quick Start Guide*. Packt Publishing, 2018.

[32] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2015.

[33] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[34] B. Krawczyk, B. T. McInnes, and A. Cano, "Sentiment classification from multi-class imbalanced twitter data using binarization," in *International Conference on Hybrid Artificial Intelligence Systems*, 2017, pp. 26–37.

[35] X. Zhou, L. Yang, X. Fan, G. Ren, Y. Yang, and H. Lin, "Self-training vs Pre-trained Embeddings for Automatic Essay Scoring," in *Information Retrieval*, 2021, pp. 155–167.

[36] W. Hidayat, E. Utami, and A. D. Hartanto, "Effect of Stemming Nazief amp; Adriani on the Ratcliff/Obershelp algorithm in identifying level of similarity between slang and formal words," in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 2020, pp. 22–27, doi: 10.1109/ICOIACT50329.2020.9331973.

[37] Y. Goldberg and G. Hirst, "Neural Network Methods in Natural Language Processing. Morgan \& Claypool Publishers(2017)," *9781627052986 (zitiert auf Seite 69)*.