

EFFECT OF WORD2VEC WEIGHTING WITH CNN-BILSTM MODEL ON EMOTION CLASSIFICATION

Merinda Lestandy¹, Abdurrahim^{2*}

¹Departement of Electrical Engineering, Faculty of Engineering, Muhammadiyah Malang University

² Informatics Master Program, Faculty of Industrial Technology, Islamic University of Indonesia

email: merindalestandy@umm.ac.id¹, 22917002@students.uin.ac.id²

Abstract

Emotion is an element that can influence human behavior, which in turn influences a decision. Human emotion detection is useful in many areas, including the social environment and product quality. To evaluate and categorize emotions derived from text, a method is required. As a result, the CNN-BiLSTM model, a classification method, aids in the analysis of the text's emotional content. A word weighting technique employing word2vec as a word weighting will help the model. The CNN-BiLSTM model with Word2vec as a pre-trained model is being used in this study to find the findings with the highest accuracy. The information is split into two groups: training and testing, and it is categorized into six categories according to how each emotion manifests itself: surprise, sadness, rage, fear, love, and joy. The best outcome from the CNN-BiLSTM model's accuracy of emotion classification is 92.85%.

Keywords : Emotion, CNN-BiLSTM, Word2Vec

Received: 12-02-2023 | **Revised:** 29-03-2023 | **Accepted:** 31-03-2023

DOI: <https://doi.org/10.23887/janapati.v12i1.58571>

INTRODUCTION

Emotion is a factor that can influence human behavior, which can impact rational tasks like making a decision or interacting with others. [1]. Emotion detection contributes to a variety of aspects that will be useful in a variety of decision-making fields, including the social environment and business [2]. Emotion recognition via text classification necessitates the use of a method that can classify the emotion of the text. Emotion analysis, which is based on computational studies of natural language expressions, seeks to identify emotions such as anger, fear, joy, sadness, and surprise [3].

Techniques for word representation in vectors are a very interesting research topic in the field of text processing or natural language processing because this topic is constantly evolving. The way word weights are represented is important because it can affect the accuracy of model learning results [4]. However, due to the unstructured nature of text, one of the difficulties in text processing is feature extraction. One of the steps in text classification is feature extraction, which converts unstructured data into structured data using a term-frequency matrix so that it can be processed using feature extraction techniques [5],[6].

The "bag of words" extraction feature, which consists of term frequency (TF), term frequency-inverse document frequency (TF-IDF),

and then n-grams, was originally used for textual data. Bag of Words, on the other hand, has the disadvantage of being unable to provide information about the meaning conveyed by the text, its structure, order, and context surrounding the words in each document. When a count-based model, such as BOW, is used, each word is counted individually, and the semantic relationships between words are not captured. If the vocabulary in the corpus is very large and the number of words in the document is very small or even non-existent, BOW can lead to overfitting because the representation of the word vector will be zero [4]. Following that, the word embedding technique was developed. Word embedding is a popular method for learning a continuous dimension by representing a vector of words, as well as a fascinating topic that is still being researched. Word embedding converts each word into a vector, with the vector representing each word in vector space.

Word2vec, a new word embedding method developed in 2013, consists of two models, skip-gram and continuous bag of words (CBOW) [7]. Word2vec, one of Mikolov's word weights, has been widely utilized for the pre-trained process because it captures the semantic sense of text by expressing text vectors that have patterns in each word that has the same meaning [8],[9]. Several studies using the word2vec

method, such as [10], show that word2vec performs well with 19,997 news articles divided into 20 topic groups derived from UCI KDD data. According to this study, the word2vec method produces good results [11]–[14]. [15] achieved an F1 value of 79% using emotion data with eight emotion labels: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. Using emotion data with labels of sadness, anger, fear, love, joy, and surprise resulted in an accuracy of 93.50%, according to [16]. With neuron trials, activation functions, and epochs, the model employs LSTM with TF-IDF (Term Frequency-Invers Document Frequency) word weighting. The accuracy of the test was 89% when compared towards the linear SVC model with the same weighting. Deep learning methods based on neural networks, such as CNN and LSTM, are currently gaining popularity due to their impressive results [17], [18]. CNNs with covolution operations can extract local features (structural patterns in images) from global information (all image data), but they are not always successful in capturing long-distance dependencies [19]. The vanishing gradient is a weakness of RNN LSTM is a component of RNN that can solve the problem of long-term dependency but cannot use text information. Despite the fact that LSTM has a close relationship with text, standard LSTM cannot effectively solve the problem. This study employs a hybrid of two deep learning models, CNN and BiLSTM, to improve and address the shortcomings of each model for emotion recognition in text [20][13].

We will research how the word2vec algorithm can be utilized to classify emotional text in this study. Word2vec will be utilized as a word weighting as a pre-trained method prior to the CNN-BiLSTM modeling process. One advantage of word2vec is that it may encode characteristics as dense vectors rather than the traditional sparse representation, which can assist in dealing with the problem of synonyms and homonyms that are common in natural language processing applications. As an outcome, the researcher proposes combining the word2vec method with the CNN-BiLSTM model.

METHOD

Obtaining classifying emotion data is the first step of this project. The data is divided into two categories: training data and testing data, with training data comprising 16000 data used to train the algorithm with the model, and testing data totaling 2000 data used to determine the performance of the model. Emotional data is separated into 6 labels (sadness, anger, fear, love, joy, and surprise).

The preprocessing procedure is used to prepare the data prior to their introduction into the method and model. This method involves lowering case, converting capital letters to lowercase letters, deleting numbers from emotional data if present, and removing punctuation. The stopwords algorithm eliminates meaningless words from emotional data by deleting the punctuation marks present in the data. Tokenization is the process of translating sentences into words, followed by weighing each word using the word2vec algorithm to determine its worth. As a classification technique, the CNN and BiLSTM combination model is used to determine the model's accuracy. Figure 1 illustrates the methodology utilized in this study. After obtaining the model classification accuracy process, the next step is to calculate the outcomes of the confusion matrix, namely accuracy, precision, and recall, which can be utilized to determine how accurate the model has been.

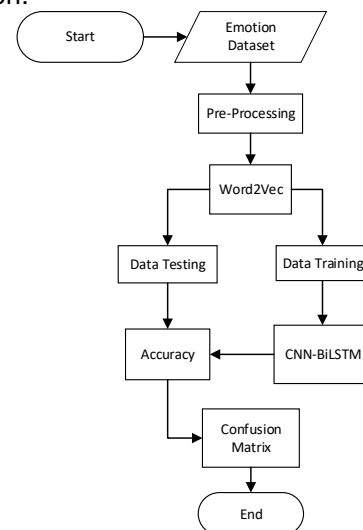


Figure 1. Research Method

Figure 2 shows several stages that begin with the emotion text data being processed into the pre-processing procedure to clean the data. Lowercase, removing numbers, removing punctuation, removing stopwords, and tokenization are the five steps. Whenever the data has been processed for word embedding segmentation, the weighting in this pre-trained model is word2vec, which converts the shape of the word into a vector, and the following phase is a modeling process whereby the data enters the convolution layer process. The max pool layer will then be connected to the fully connected layer as a whole, and the bi-LSTM process will begin. The sigmoid activation function is utilized, and the end outcome of emotion text is accuracy.

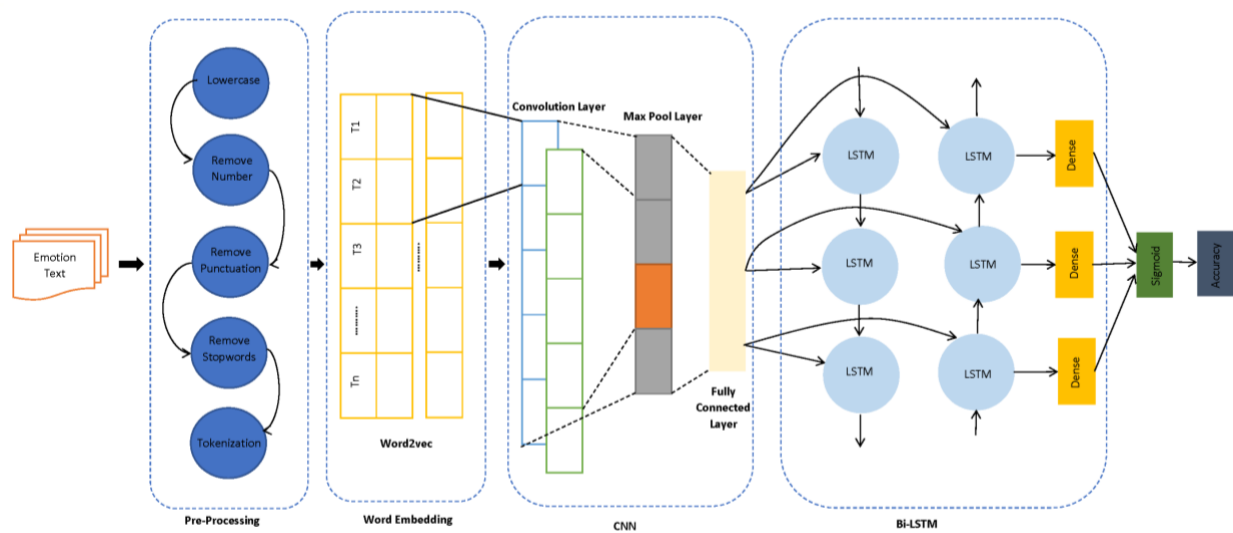


Figure 2. Proposed Method

Dataset

The initial phase in this research is to gather data and emotions for the classification process. The data is divided into two parts: testing data and training data, which is useful for preventing the overfitting process from occurring during the training process. The emotion dataset can be found at <https://www.kaggle.com/datasets/praveengovi/e>

motions-dataset-for-nlp. Approximately 16000 training data and 2000 testing data are used, according to research [15]. The information is derived from the Twitter API and includes multiple types of emotions, including anger, a description of "angry" and "pissed," fear, a description of "fear" and "worried," and pleasure, a description of "fun" and "joy," among others.

Table 1. Emotion Dataset

Text	Label
i started feeling sentimental about dolls i had as a child and so began a collection of vintage barbie dolls from the sixties	Sadness
i feel a bit rude writing to an elderly gentleman to ask for gifts because i feel a bit greedy but what is christmas about if not mild greed	Anger
im getting the feeling that my classes are a little intimidated by the concept of a lit	Love
i was ready to meet mom in the airport and feel her ever supportive arms around me	Fear
i have immense sympathy with the general point but as a possible proto writer trying to find time to write in the corners of life and with no sign of an agent let alone a publishing contract this feels a little precious	Joy
i am now nearly finished the week detox and i feel amazing	Surprise

Preprocessing Data

Before entering the weighting procedure and model, the preprocessing step is where data is prepared. In this procedure, there are five preprocessing stages: lower case, removal of numerals, removal of punctuation, removal of stopwords, and tokenization. In this study, the preprocessing technique employs a number of steps, including:

a. Using Lower Case The feature's primary function is to convert all letters in a word to lowercase [21].

- b. Remove Number: Since the number contained in the emotion data has no effect on the classification results, deleting it can minimize noise and improve productivity [22].
- c. Remove Punctuation; punctuation refers to distinctive characters like as exclamation marks, commas, and question marks that are not necessary for data classification [23].
- d. Remove stopwords; words such as "a," "an," and "the" have no sense in the text and can be removed [24], [25].

- e. Tokenization is the separation of words into word or token form, with each token divided into words, phrases, or paragraphs [26].

Word2Vec

Word2Vec is one of the strategies used to distribute a word representation further. This method is divided into two components CBOW and skip-gram architecture, which are important for determining a word's representation vector. CBOW predicts the target word based on the results of the source context, whereas Skip-Gram predicts the word based on the word in the source context of the target word. Both variants are beneficial for assisting students in learning similar words based on textual words [27].

Pretrained Word2Vec

This method employs pre-trained Wikipedia2Vec data from the word weighting method, with dimensions of 100 and 300. This enwiki has been configured to utilize the values window = 10, iteration = 10, and negative = 15.

Convolutional Neural Network

CNN is one of the algorithms for deep learning that uses images as input. CNN employs a convolution method, specifically a matrix that extracts features from the input matrix. Pixels or images in a text or document are substituted with character data during CNN natural language processing, where the text data to be processed is the matrix of the input data

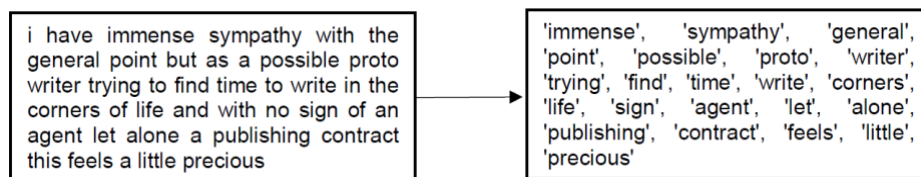


Figure 3. Preprocessing Result

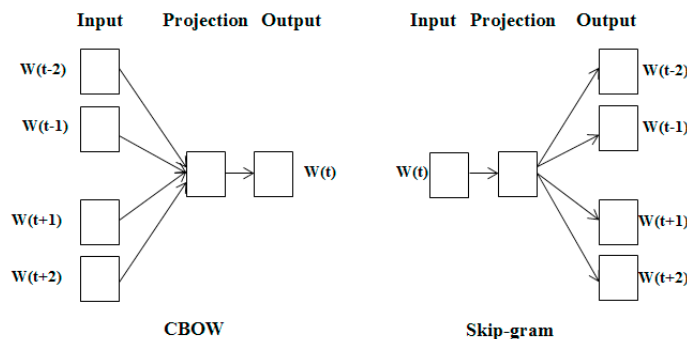


Figure 4. Word2Vec Structure[7]

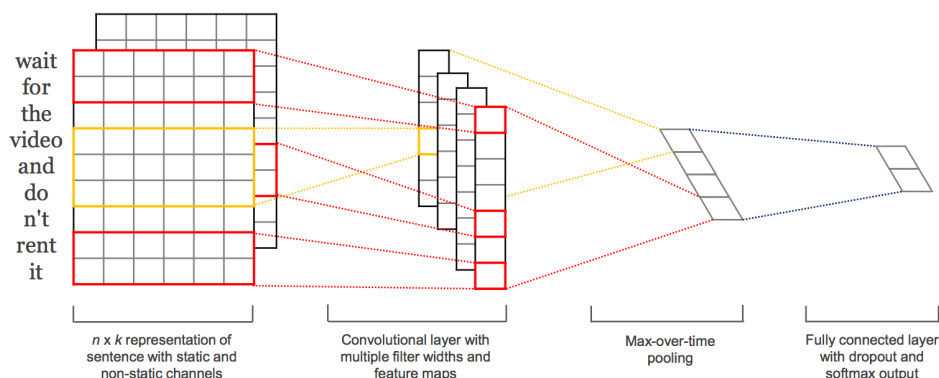


Figure 5. CNN Architecture [19]

Bidirectional Long Short Term Memory

Text classification is performed progressively using a recurrent neural network (RNN) in the LSTM unit. RNN can recall and store information from previous texts, but it has the disadvantage

of being unable to remember and keep information for an extended period of time due to its limited storage capacity. LSTM, a development model for RNN, was created to overcome the model's flaws.

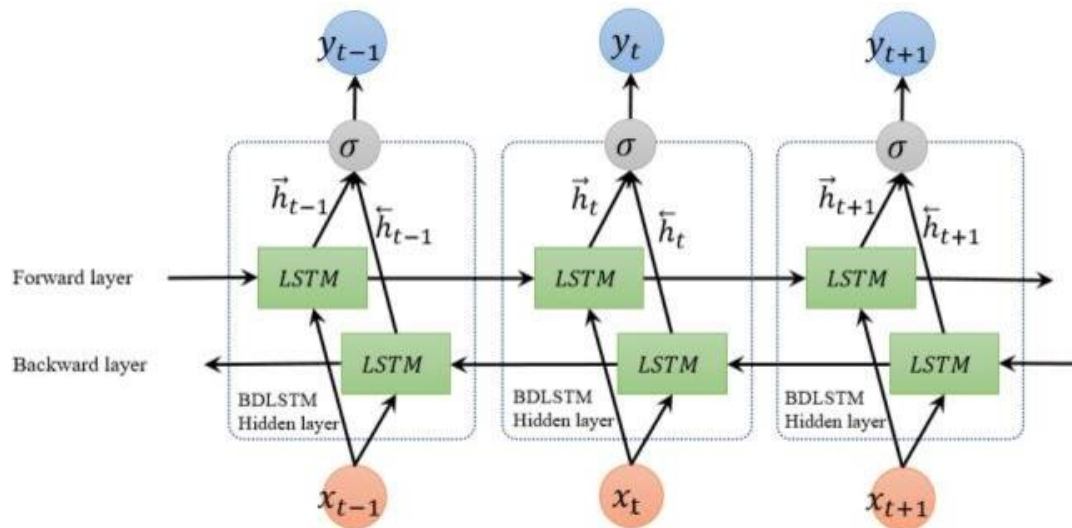


Figure 6. BiLSTM Architecture [28]

The image above shows the Bi-LSTM architecture. This model is the evolution of LSTM with two layers whose processes operate in the reverse manner. Because each word is evaluated sequentially, this model is particularly effective at spotting text patterns inside phrases. The forward layer processes from the first to the

final word, whereas the backward layer processes from the last to the first word. Thus, there are layers with two opposing directions so that the model can comprehend the preceding word and the leading word, so that the model's process will be deeper, enabling it to comprehend the text's context more thoroughly.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 35, 100)	1506500
dropout (Dropout)	(None, 35, 100)	0
conv1d (Conv1D)	(None, 33, 32)	9632
max_pooling1d (MaxPooling1D)	(None, 11, 32)	0
dropout_1 (Dropout)	(None, 11, 32)	0
conv1d_1 (Conv1D)	(None, 9, 32)	3104
max_pooling1d_1 (MaxPooling1D)	(None, 3, 32)	0
dropout_2 (Dropout)	(None, 3, 32)	0
bidirectional (Bidirectional)	(None, 512)	591872
dropout_3 (Dropout)	(None, 512)	0
dense (Dense)	(None, 256)	131328
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 6)	1542

Total params: 2,243,978
Trainable params: 2,243,978
Non-trainable params: 0

Figure 7. Embedding Size 100

```

Model: "sequential"
Layer (type)                Output Shape                Param #
-----
embedding_2 (Embedding)     (None, 35, 300)            4519500
dropout (Dropout)           (None, 35, 300)            0
conv1d (Conv1D)              (None, 33, 32)             28832
max_pooling1d (MaxPooling1D) (None, 11, 32)             0
dropout_1 (Dropout)          (None, 11, 32)             0
conv1d_1 (Conv1D)            (None, 9, 32)              3104
max_pooling1d_1 (MaxPooling1 (None, 3, 32)             0
dropout_2 (Dropout)          (None, 3, 32)             0
bidirectional (Bidirectional (None, 512)             591872
dropout_3 (Dropout)          (None, 512)                0
dense (Dense)                (None, 256)                131328
dropout_4 (Dropout)          (None, 256)                0
dense_1 (Dense)              (None, 6)                  1542
-----
Total params: 5,276,178
Trainable params: 5,276,178
Non-trainable params: 0
  
```

Figure 8. Embedding Size 300

According to the image above, the layers employed are 35 layers with embedding sizes of 100 and 300. The maximum number of pooling layers is 32, the bidirectional layer is 512, and the density is 6. Several experiments were run in this study to determine the ideal value by varying the

values of the parameters. The parameters that have been adjusted are the embedding size, batch size, and optimizer with a learning rate of 0.001. RMSprop and Adamax were the optimizers employed in this study.

Table 2. Neuron Test

Neuron	Test Accuracy
32	91.40%
64	91.15%
128	91.65%
256	92.00%
512	92.30%

Several experiments on the neuron layer were performed in the table. The numbers 32, 63, 128, 256, and 512 are used to alter the neurons five times. This test is useful for determining the best

value for the classification model displayed in the accuracy rate test. The best results are obtained with 512 neurons and a test accuracy score 92.30%.

Table 3. Activation Function Test

Neuron	Activation Function	Accuracy	Loss
512	Softmax	99.40%	0.0628
512	Sigmoid	99.43%	0.0660

Table 4. Word2Vec Result

Epoch	Embedding Dimension	Batch Size	Optimizer	Test Accuracy
30	100	128	RMSprop	92.50%
		128	Adamax	90.15%
	300	128	RMSprop	92.85%
		128	Adamax	91.90%

Table 5. Test results obtained without Word2Vec

Epoch	Dimension	Batch Size	Optimizer	Test Accuracy
30	100	128	RMSprop	86.15%
		128	Adamax	77.70%
	300	128	RMSprop	86.80%
		128	Adamax	82.05%

The model's embedding dimension and optimizer are changed in the final test experiment. The results of using 512 neurons and the sigmoid function are displayed in the table; the best accuracy of word weighting applying word2vec is 92.85% with embedding size 300 and optimizer RMSprop. Pada tabel 5 juga melakukan pengujian tanpa menggunakan pembobotan kata, pengujian paramet pada tabel 5 sama dengan pengujian pada tabel 4 yakni perubahan dari optimizer serta pada tabel 5 tidak menggunakan pembobotan kata word2vec. Terlihat bahwa hasil akurasi tanpa menggunakan word2vec, akurasi yang dihasilkan paling tinggi sebesar 86.80%.

Tables 4 and 5 show that applying the word2vec word weighting method boosted accuracy, as seen by the table 4 experiments, where the maximum accuracy reached 92.50%. This is due to word2vec's ability to extract semantic and syntactic information from words. When Word2vec learns a word representation, each word begins at a random point in the vector space. Following that, the word will be gradually

moved to a place closer to a word with familiarity or similarity to the location of similar words depending on their neighbors in the training data. In contrast to Table 5, the greatest accuracy value achieved utilizing test data without word weighting is 86.80%.

CONCLUSION

The CNN-BiLSTM combination model has various tests, including neuron testing with an ideal value of 92.30%, activation functions resulting in 99.43% accuracy, as well as RMSprop and Adamx optimizers, based on research including emotion data with word weighting of word2vec. Each test embedding size of 100 provides the highest test accuracy of 92.50% when using the RMSprop optimizer, while 300 gives the most ideal accuracy of 92.85% when applying the RMSprop optimizer. The optimizer achieved the highest CNN-BiLSTM model performance with a Word2Vec weighting of 92.85% and an embedding dimension of 300.

REFERENCES

- [1] M. A. M. Shaikh, H. Prendinger, and M. Ishizuka, "A Linguistic Interpretation of the OCC Emotion Model for Affect Sensing from Text BT - Affective Information Processing," J. Tao and T. Tan, Eds. London: Springer London, 2009, pp. 45–73. doi: 10.1007/978-1-84800-306-4_4.
- [2] F. Fanesya, R. C. Wihandika, and Indriati, "Deteksi Emosi pada Twitter Menggunakan Metode Naive Bayes dan Kombinasi Fitur," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 7, p. 3, 2019.
- [3] A. Bandhakavi, N. Wiratunga, D. Padmanabhan, and S. Massie, "Lexicon based feature extraction for emotion text

- classification," *Pattern Recognit. Lett.*, vol. 93, pp. 133–142, 2017, doi: 10.1016/j.patrec.2016.12.009.
- [4] 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.
- [5] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining Cambridge*, vol. 2001. 2001. [Online]. Available: <http://link.springer.com/10.1007/978-1-4471-4884-5>
- [6] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. USA: Cambridge University Press, 2008.
- [7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.
- [8] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2vec model analysis for semantic similarities in English words," *Procedia Comput. Sci.*, vol. 157, pp. 160–167, 2019, doi: 10.1016/j.procs.2019.08.153.
- [9] D. I. Af'idah, R. Kusumaningrum, and B. Surarso, "Long Short Term Memory Convolutional Neural Network for Indonesian Sentiment Analysis towards Touristic Destination Reviews," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2020, pp. 630–637. doi: 10.1109/iSemantic50169.2020.9234210.
- [10] E. M. Dharma, F. L. Gaol, H. L. H. S. Warnars, and B. Soewito, "the Accuracy Comparison Among Word2Vec, Glove, and Fasttext Towards Convolution Neural Network (Cnn) Text Classification," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 2, pp. 349–359, 2022.
- [11] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews," *Procedia Comput. Sci.*, vol. 179, no. 2020, pp. 728–735, 2021, doi: 10.1016/j.procs.2021.01.061.
- [12] A. K. Sharma, S. Chaurasia, and D. K. Srivastava, "Sentimental Short Sentences Classification by Using CNN Deep Learning Model with Fine Tuned Word2Vec," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 1139–1147, 2020, doi: 10.1016/j.procs.2020.03.416.
- [13] W. Yue and L. Li, "Sentiment analysis using word2vec-cnn-bilstm classification," *2020 7th Int. Conf. Soc. Netw. Anal. Manag. Secur. SNAMS 2020*, pp. 3–7, 2020, doi: 10.1109/SNAMS52053.2020.9336549.
- [14] L. Xiao, G. Wang, and Y. Zuo, "Research on Patent Text Classification Based on Word2Vec and LSTM," *Proc. - 2018 11th Int. Symp. Comput. Intell. Des. Isc. 2018*, vol. 1, pp. 71–74, 2018, doi: 10.1109/ISCID.2018.00023.
- [15] E. Saravia, H. T. Liu, Y. Huang, J. Wu, and Y. Chen, "CARER: Contextualized Affect Representations for Emotion Recognition," pp. 3687–3697, 2018.
- [16] M. I. Alfarizi, L. Syafaah, and M. Lestandy, "Emotional Text Classification Using TF-IDF (Term Frequency-Inverse Document Frequency) And LSTM (Long Short-Term Memory)," *JUITA J. Inform.*, vol. 10, no. 2, p. 225, 2022, doi: 10.30595/juita.v10i2.13262.
- [17] D. S. Ashari, B. Irawan, and C. Setianingsih, "Sentiment Analysis on Online Transportation Services Using Convolutional Neural Network Method," in *2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Oct. 2021, pp. 335–340. doi: 10.23919/EECSI53397.2021.9624261.
- [18] M. Lestandy, A. Abdurrahim, and L. Syafa, "Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent," vol. 5, no. 10, pp. 802–808, 2021.
- [19] Y. Kim, "Convolutional Neural Networks for Sentence Classification," 2014, doi: 10.48550/ARXIV.1408.5882.
- [20] K. Zhou and F. Long, "Sentiment Analysis of Text Based on CNN and Bi-directional LSTM Model," *2018 24th Int. Conf. Autom. Comput.*, no. September, pp. 1–5, 2018, doi: 10.23919/IConAC.2018.8749069.
- [21] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Inf. Process. Manag.*, vol. 50, no. 1, pp. 104–112, 2014, doi: 10.1016/j.ipm.2013.08.006.
- [22] M. Khader, A. Awajan, and G. Al-Naymat, "The Effects of Natural Language Processing on Big Data Analysis: Sentiment Analysis Case Study," in *2018 International Arab Conference on Information Technology (ACIT)*, 2018, pp. 1–7. doi: 10.1109/ACIT.2018.8672697.
- [23] A. Filcha and M. Hayaty, "Implementasi

- Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa," *JUITA J. Inform.*, vol. 7, no. 1, p. 25, 2019, doi: 10.30595/juita.v7i1.4063.
- [24] B. R. Savaliya and C. G. Philip, "Email fraud detection by identifying email sender," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 1420–1422. doi: 10.1109/ICECDS.2017.8389678.
- [25] S. Pradha, M. N. Halgamuge, and N. Tran Quoc Vinh, "Effective text data preprocessing technique for sentiment analysis in social media data," *Proc. 2019 11th Int. Conf. Knowl. Syst. Eng. KSE 2019*, pp. 1–8, 2019, doi: 10.1109/KSE.2019.8919368.
- [26] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. A. Elaziz, and A. Dahou, "A Study of the Effects of Stemming Strategies on Arabic Document Classification," *IEEE Access*, vol. 7, pp. 32664–32671, 2019, doi: 10.1109/ACCESS.2019.2903331.
- [27] W. López, J. Merlino, and P. Rodríguez-Bocca, "Learning semantic information from Internet Domain Names using word embeddings," *Eng. Appl. Artif. Intell.*, vol. 94, p. 103823, 2020, doi: <https://doi.org/10.1016/j.engappai.2020.103823>.
- [28] H. F. Fadli and A. F. Hidayatullah, "Identifikasi Cyberbullying pada Media Sosial Twitter Menggunakan Metode LSTM dan BiLSTM," *Uij*, vol. 2, no. No.1, pp. 1–6, 2021, [Online]. Available: <https://journal.uii.ac.id/AUTOMATA/article/view/17364>