Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Combining Bi-LSTM And Word2vec Embedding For Sentiment Analysis Models of Application User Reviews

Verra Budhi Lestari*, Ema Utami, Hanafi
verrabl@students.amikom.ac.id, ema.u@amikom.ac.id, hanafi@amikom.ac.id
Universitas Amikom Yogyakarta

Article Information

Submitted : 30 Dec 2023
Reviewed: 3 Jan 2024
Accepted : 10 Feb 2024

Keywords

Sentiment Analysis, LSTM, Bi-LSTM, Dense Layer, Word2Vec

Abstract

The objective of application and product sentiment analysis is to determine the polarity of sentiment in application reviews based on user feedback. The LSTM model and its derivatives, such as the GRU, are becoming increasingly popular among the various Neural Network designs employed in sentiment analysis. However, the LSTM model continues to have issues with how slowly it converges and can only gather data in a single direction, so in order to enhance performance, other deep learning techniques like Bi-LSTM must be used. In addition, word embedding combinations can be taken into consideration in order to generate an evergreater variety of word representations. This research presents a comparison of LSTM and BiLSTM models in order whether the performance of the combination of the BiLSTM model with the application of word embedding is capable of having better performance than the one-way LSTM model. The accuracy test results demonstrated that BiLSTM model with Word2Vec gets the highest accuracy test results, with an accuracy of 87%. This proves that the BiLSTM model with Word2Vec is able to have better performance than LSTM.

A. Introduction

Sentiment analysis has gained popularity as a study topic in the domains of data mining and natural language processing in the last decade [1] [2]. Sentiment analysis is one of Natural Language Processing's primary functions (NLP). Once attitudes, ideas, opinions, or judgments regarding a given topic have been retrieved, sentiment recognition can be useful for governments, corporations, and individual decision-makers [3]. Text documents are categorized using sentiment analysis into positive, neutral, and negative sentiment groups. Public opinion on a subject can be ascertained by categorizing people into these groups according to their perceptions [4]. Text sentiment analysis can also be used to categorize an individual's emotions [5].

Sentiment analysis covers a wide range of subjects, including cyberbullying [6], identifying emotions from social media expressions [7], and phenomena like Covid19 and the Covid19 Vaccine [8]. In addition, sentiment analysis examines opinions that individuals have expressed in text reviews, including reviews of movies [9] [10], apps reviews [11] [12] [13], drugs [14], and marketplace products [15] [16] [17] [18].

Finding the polarity of sentiment in application reviews with a lot of data is the goal of application and product sentiment analysis. Sentiment analysis results are used to learn what users think of the application, how the public perceives it, and what features the developer should add-all of course depending on user feedback [2] [15]. Comparable to a product, an application needs to meet user needs in order to compete with other comparable applications. This means that applications need to be developed. Users can share their thoughts about an application's quality, features, services, and appearance by writing reviews on application provider platforms like Google Play Store, App Store, Amazon, and others [19].

Even though there is a rating feature available on the Play Store, this feature is considered less accurate, this is because in some ratings there is a discrepancy between the review text and the rating given, besides that not all ratings have review text. That way we only use ratings that have review text, then utilize sentiment analysis techniques to analyze the collection of customer reviews. Sentiment analysis of the user review data will be utilized to inform development decisions for the application [19].

Long Short-term Memory (LSTM) model and its variations, including the Gated Recurrent Unit (GRU), are drawing more and more attention among the several Neural Network architectures used for sentiment analysis [1]. On the other hand, the LSTM model continues to have issues with how slowly it converges. Additionally, it can only gather data in a single direction, so in order to enhance performance, other deep learning techniques like BiLSTM must be used [6] [7] [19]. The inquiry pertains to whether the LSTM architecture's built-in gates are currently capable of making accurate clasification. By navigating through the input data in two directions, BiLSTMs allow for additional training.

Aside from that, word embedding combinations can be taken into consideration in order to generate an evergreater variety of word representations [12] [17]. The application of Word2Vec was carried out because the combination of the LSTM deep learning model with Word2Vec word embedding feature extraction to carry out sentiment analysis showed that this model was able to excel compared to Naïve Bayes, Support Classification, and CNN in previous research [19].

Then, the key research topic is whether BiLSTM performs better than standard unidirectional LSTM with its extra training capabilities. This research presents a comparison of LSTM and BiLSTM models. Threfore in order whether the performance of the combination of the BiLSTM model with the application of word embedding is capable of having better performance than the one-way LSTM model, this research will combine the BiLSTM model with Word2Vec to carry out sentiment analysis on user reviews of the Gojek application based on user reviews written in Indonesian. This research will assess the performance of the algorithm using accuracy, precision, recall, and F1 measure.

B. Research Method

Sentiment analysis process for application user reviews presented in Figure 1, it begins by collecting a public dataset. This dataset has labels for each sentiment category, namely reviews with positive and negative sentiment. Before the data set is applied to the model, the dataset is divided into 2 parts, namely 80% training data and 20% test data [20] [21]. Several preprocessing stages were also carried out such as clean text, stopword removal, stemming, and tokenization which will be mentioned in the discussion of Preprocessing Data. After going through the preprocessing stages, the data. This will be drilled and tested by applying extraction features in the form of word embedding Word2Vec, then experiments will be carried out by applying several sentiment analysis models such as LSTM, BiLSTM, and the BiLSTM-Dense Layer combination. The results of the sentiment analysis classification will be evaluated or validated using accuracy, precision, recall and F1-Measure until results are obtained in the form of a model that has the best performance when combined with Word2Vec to analyze the sentiment of Gojek application user reviews.

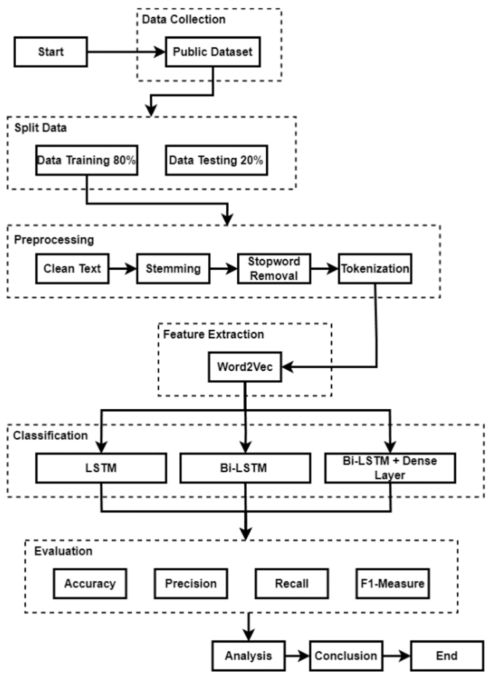

Figure 1. Research Flow

1. Dataset

This study's raw material was the Gojek User Reviews dataset, which is available on public (Github.com). For the purposes of this study, the dataset will be split into two categories: test data and training data. Up to 80% of the training data is used to teach the algorithm to identify the appropriate model. While the Test Data included up to 20% of newly collected data without a class yet, a classification procedure is still required to identify the appropriate class. The class (label) of the data obtained is positive and negative.

2. Preprocessing

Cleaning

Clean Text is a method for characterizing text and extracting text from data that isn't used in further analysis, such as removing or obscuring text fields, numeric, symbols, hyperlinks, hash tags, emoticon, special character, username, and unused data columns [7] [15].

Stemming

Despite the fact that many words in natural language are identical, recognition is worthless due to their varied forms. Stemming is the process of removing suffixes and prefixes from an input word in order to recover the parent (root) of a phrase, which all associated words will share. For example, the same root will be used for computations and stemming often results in root words that are invalid or irrelevant [7]. Reducing word variations with almost identical meanings in a document is the goal of stemming, which aims to enhance and streamline the information search process [13]. The "Sastrawi" stemming method was used in this study.

Stopword Removal

Stop words removal constitute a process where unimportant words in a text document will be removed. These include High-frequency words such as subjects, affixes, pronouns, determinants, prepositions, conjunctions, and others are also known as function words or closed-class words [9]. "Indonesian" is the stopword used in this study.

Tokenization

Tokenization is the process of identifying words within the input sequence of characters, mostly by identifying and separating punctuation marks, abbreviations, and other symbols. The tokenization process may also involve steps for normalizing the text, such as eliminating HTML tags, username, true casing, or lowercase lettering [9].

SMOTE

A method of oversampling where the minority class is oversampled without replacing the samples, but rather by creating "synthetic" cases. They used specific actions on real data to create more training data. Each minority class sample is taken, then synthetic examples are introduced along the lines segments joining any/all of the minority class nearest neighbors, oversampling the minority class. Selecting neighbors at random from the nearest neighbors depends on the degree of over-sampling needed. For example, just two neighbors are selected from the five closest neighbors and one sample is generated in each direction if 200% over-sampling is required.

The following method is used to generate synthetic samples: Calculate the difference between the sample under examination, or the feature vector, and its closest neighbor. Add this difference to the feature vector under consideration by multiplying it by a random value between 0 and 1. This results in the random point being selected along the line segment that connects two particular features. This strategy essentially compels the minority class's decision-making region to become more inclusive [22].

3. Word2Vec

Word embedding is a method of representing each word numerically and vectorically by transforming the fundamental word into a vector form [23]. Word embedding is part of feature extraction; certain word embeddings suggest related ideas or methods [24].

In 2013, Mikolov. introduced Word2Vec as one of the word embedding techniques. Word2Vec is able to determine the meaning of words that are similar to the target word by focusing on terms that are near to it. The Word2Vec approach is highly popular because of its benefits. Two methods are used by Word2Vec: the skip-gram model and the continuous word collection methodology [7].

The number of words in the corpus multiplied by the number of hidden neurons in the hidden layer determines the dimensions of the weight matrix in each layer [25]. Words are converted to vectors using the weight matrix located in the drilled hidden layer. Similar to a lookup table, this weight matrix has rows representing individual words and columns representing word vectors. Words can be represented as vectors in semantic space by using this weight matrix [26].

4. Sentiment Analysis

Sentiment analysis, which is commonly referred to as opinion gather-ing, encompasses a number of tasks, such as sentiment extraction, sentiment classi-fication, subjectivity categorization, opinion summarization, and written opinion spam detection. Inquiries about feelings, attitudes, and opinions about a range of topics, such as businesses, people, events, issues, products, subjects, organizations, and services, are the goal of this activity. Examining their beliefs and attitudes is another goal [27] [28].

LSTM

RNNs struggle to learn long-term dependencies, as was previously mentioned. The vanishing gradient problem can be effectively addressed by RNNs, and LSTM-based models are an extension of these models. In order to allow RNNs to retain and learn long-term dependencies on inputs, the LSTM models essentially expand their memory. With the help of this memory extension, users will be able to read, write, and erase information from their memories for extended periods of time. Because the LSTM memory has the option to choose whether to preserve or ignore memory information, it is referred to as a "gated" cell. An LSTM model is able to extract significant features from inputs and retains this data for an extended amount of time [29].

Based on the weight values assigned to the information throughout the training process, the decision is made regarding whether to delete or preserve the information. As a result, an LSTM model learns what data is important to keep or discard [29]. Three gates make up an LSTM model in general: forget, input, and output gates. The output gate determines whether the current value in the cell contributes to the output, the forget gate decides whether to keep or remove the existing information, and the input gate determines how much new information will be added to the memory [30] [31].

BiLSTM

Contextual information is captured using BiLSTM, which combines an LSTM unit with a bidirectional recurrent neural network (BiRNN) model. The BiLSTM model handles all inputs in the same way, but in sentiment analysis, the words that contain sentiment information have a significant impact on the text's sentiment polarity. To solve the issue, this approach presents an improved word sentiment vector [32].

By applying two LSTMs to the input data, the bidirectional LSTM are an extension of the previously mentioned LSTM models. An LSTM (i.e., forward layer) is applied to the input sequence in the initial round, the input sequence is sent into the LSTM model (also known as the backward layer) in reverse form during the second round [33], as seen in Figure 2. By using the LSTM twice, long-term dependencies are better learned, which increases the accuracy of the model [34].

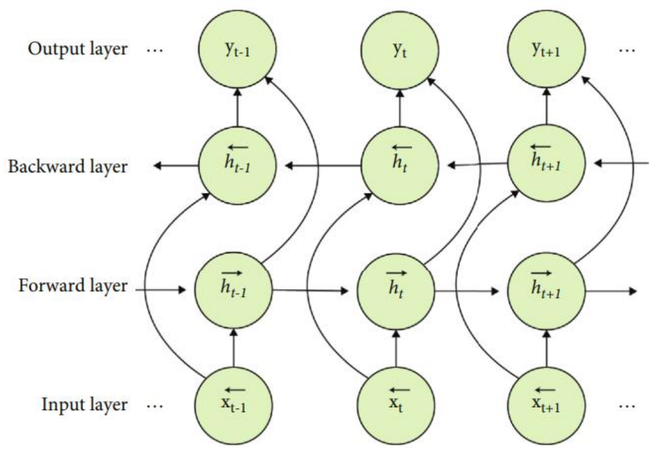

Figure 2. Bidirectional LSTM [8]

5. Evaluation Matrix

Confussion Matrix is used to examine how a model's work is related to several labels. Confusion matrix illustrates Model categorization performance and how accurately a model predicts particular situations [18]. Accuracy, precision, recall, and F-measure are the metrics that are most commonly employed in text classification. As each metric has a value between 0 and 1, the objective is to maximize them all. As a result, higher numbers represent better success in classification. Confussion Matrix that presented in Table 1 provides useful understanding of true negative (TN), false positive (FP), false negative (FN), and false positive (FP) [35].

Table 1. Confusion Matrix [36]

	Predict Class	
Actual Class	**Positive**	**Negative**
Positive	True Positive	False Negative
Negative	False Positive	True Negative

Categorization performance is measured using an accuracy metric. Accuracy is determined by the number of samples that have been identified appropriately. Accuracy is employed when the classification reliably predicts a class, which may be ascertained using a formula like equation 1.

Precision and Recall are two metrics that are commonly combined to evaluate the retrieval effectiveness of information and are commonly used statistics in text classification. More precisely, precision and recall measure the number of relevant documents that were successfully retrieved and the formula is found in Equations 2 and Equations 3 can be used to determine both metrics.

Recall and accuracy are rarely taken into account separately. F-measure sometimes combines these two measurements, providing a single weighted statistic for evaluating overall perfor-mance. Equation 4 can be used to compute the F-measure by utilizing the formula in the equation.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP}\ x\ 100\% \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$f - maeasure = \frac{2*Recall*Precision}{Recall+Precision} \tag{4}$$

C. Result and Discussion

The method used for sentiment analysis in this study is using the Long Short-term Memory, Bidirectional LSTM, and CNN with Word2Vec Word Embedding.

1. Data Collecting

At the phase of data collection, the author examines information from multiple sources, one of which is accessible via the public dataset platform, Github.com. The author was able to collect a dataset comprising 1,782 user reviews of Gojek from the Playstore. As seen in Figure 3, the dataset consists of 721 positive reviews and 1,061 negative reviews.

This research was carried out using 1,782 review data, dividing the data into training data and test data in a ratio of 80:20 [20] [21], namely 1,603 review data for training data and 179 review data for test data.

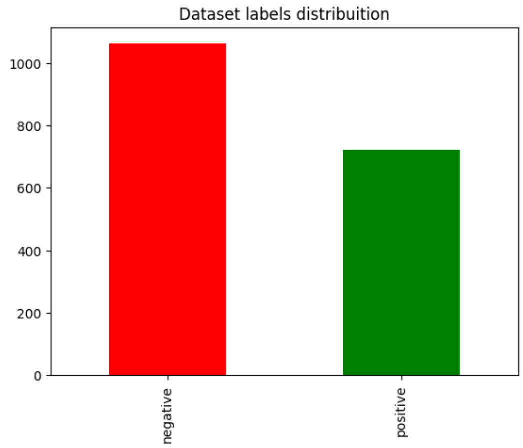

Figure 3. Dataset Distribution

2. Implementation

To achieve the best findings, we implemented several experiment using various parameters. Wordcloud both preprocesing and after preprocessing, followed by Word2Vec embedding, were the parameters that were examined. Wordcloud without preprocessing is the parameter being tested for the first one. It processes all the data, which comes to 249.277 words, in this wordcloud. These are the terms that were in the wordcloud in Figure 4.a prior to preprocessing.

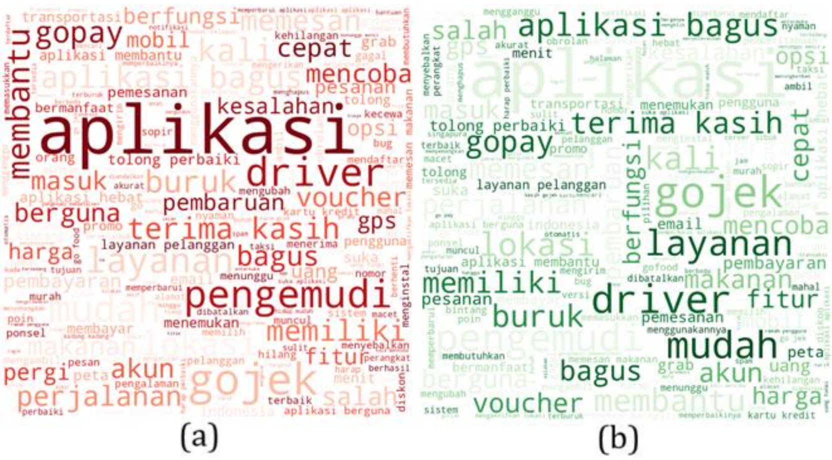

Figure 4. Wordcloud **(a)** Before Preprocessing; **(b)** After Preprocessing

We have extracted the subject words, conjunctions, and typos from the 249.277 words of data after processing them all. This allows the preprocessing system to read the processed words. In addition, we used the helpful tqdm package to show a progress bar with a basic loop depending on the results of repeated activities, like looping through a set of data. Additionally, the wordcloud following preprocessing is the second parameter evaluated. Process data that has already been processed before preprocessing during this step. There are 125.804 words in this preprocessing. These are the preprocessed terms from the wordcloud in Figure 4.b. The feature extraction and evaluation matrix are then followed.

Among the oversampling techniques most frequently employed to address the imbalance issue is SMOTE (Synthetic Minority Oversampling Technique) [37]. By combining preexisting minority instances, SMOTE creates new minority instances. By using linear interpolation, it creates the virtual training records for the minority class [38]. In this study, with sampling_strategy=1, each example in the minority class has one or more k-nearest neighbors that are randomly chosen to create these synthetic training records. Following the oversampling procedure, the data is rebuilt, and the modified data can be subjected to multiple classification models. In the training set, our distribution was negative at 959 and positive at 644. Following SMOTE, it changes to 959 (negative), 959 (positive).

3. Evaluation Model

Based on the analysis and discussion of the experiments that have been carried out, it can be stated that LSTM, Bidirectional LSTM, and CNN utilizing Word2Vec embedding with the Gojek user review data for this study have been successfully implemented. The preprocessing procedure for filtering and classifying data using LSTM, Bidirectional LSTM, and CNN was finished with 1,782 data for the purpose this study. Up to 80% (1,603 reviews) for training data and 20% (179 reviews) for test data were aggregated, 721 positive data and 1,061 negative data make up the opinion. Where to avoid the problem of overfitting in conditions of unbalanced data, the process of handling the training data is carried out by creating synthetic samples from minority classes using the SMOTE method.

Based on the experience of word2vec embedding in the LSTM model, the classification results were obtained as shown in the confusion matrix of Figure 5 are as follows: true positives are 62, true negatives are 81, false positives are 15, and false negatives are 21. The results of the model performance are also presented in Table 2.

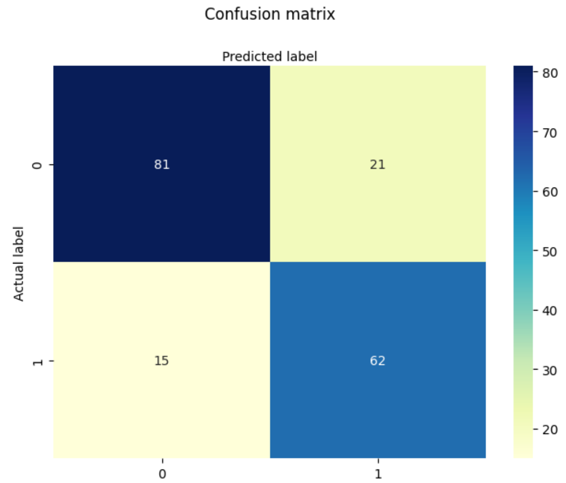

Figure 5. Confusion Matrix LSTM + Word2Vec

Table 2. Result of LSTM + Word2Vec

LSTM + Word2Vec		
	Positive	**Negative**
Accuracy	0.80	
Precision	0.75	0.84
Recall	0.81	0.79
F-measure	0.77	0.82

Afterwards, the word2vec embedding on the BiLSTM model yields classification results were obtained as shown in the confusion matrix of Figure 6 with true positives of 59, true negatives of 96, false positives of 18, and false negatives of 6. Table 3 also displays the results of the model's performance.

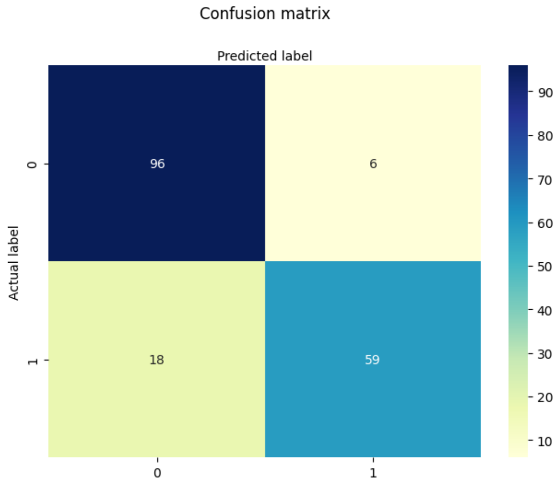

Figure 6. Confusion Matrix BiLSTM + Word2Vec

Table 3. Result of BiLSTM + Word2Vec

BiLSTM + Word2Vec		
	Positive	**Negative**
Accuracy	0.87	
Precision	0.91	0.84
Recall	0.77	0.94
F-measure	0.83	0.89

Then, using a Dense Layer combination to implement word2vec embedding in the BiLSTM model as shown in the confusion matrix of Figure 7, the classification results are as follows: true positives are 59, true negatives are 86, false positives are 18, and false negatives are 16. The results of the model's execution are also shown in Table 4.

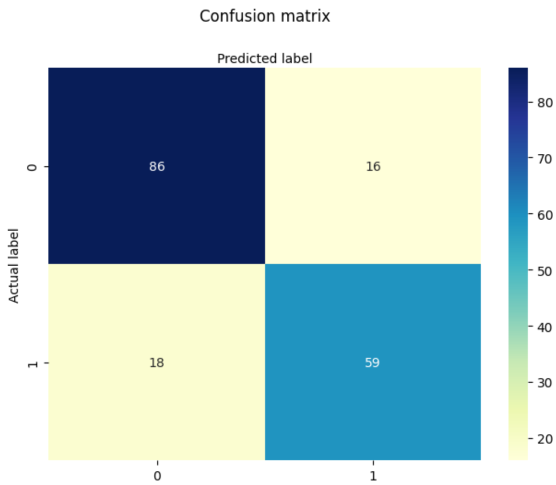

Figure 7. Confusion Matrix BiLSTM + Dense Layer + Word2Vec

Table 4. Result of BiLSTM + Dense Layer + Word2Vec

BiLSTM + DenseNet + Word2Vec		
	Positive	**Negative**
Accuracy	0.81	
Precision	0.79	0.83
Recall	0.77	0.84
F-measure	0.78	0.83

The performance outcomes of applying the Word2Vec embedding extraction feature to each LSTM, BiLSTM, and BiLSTM + Dense Layer model were then analyzed, as shown in Table 5. For Word2Vec + LSTM, Word2Vec + BiLSTM, and Word2Vec + BiLSTM-Dense Layer, the performance gains are as follows: Accuracy 80%, 87%, 81%; Precision 79.5%, 875%, 81%; Recall 80%, 85.5%, 80.5%; and f-measure 79.5%, 86%, 80.5%. According to the findings, when it comes to doing sentiment analysis on user reviews of the Gojek application, the BiLSTM model with Word2Vec word embedding performs better than the LSTM and BiLSTM-Dense Layer models.

Table 5. Comparison Evaluation Result

	Evaluation			
	Accuracy	**Precision**	**Recall**	**F1-Score**
LSTM	0.800	0.795	0.800	0.795
BiLSTM	**0,870**	**0.875**	**0.855**	**0.860**
BiLSTM + DenseNet	0,810	0.810	0.805	0.805

By processing data in two directions, BiLSTM can assist solve the vanishing gradient problem and lower the likelihood of losing information at the start or conclusion of the text, which is why it is seen to be superior. When context from both sides is required, like in sentiment analysis jobs where a word's meaning depends on its preceding and following words, BiLSTM functions well. It processes text simultaneously from beginning to finish and from end to beginning. Furthermore, sequence modeling and intricate context-dependent context understanding are two particular tasks where BiLSTM performs better. This gives each word a deeper and more comprehensive context that the model can access. LSTM, on the other hand, is thought to be less effective in managing reverse context because it processes text only from the beginning and typically only in one direction, which causes it to lose some contextual information at the end of a sentence or phrase in sentiment analysis.

D. Conclusion

It is possible to take the results of the analysis and discussion as meaning that the LSTM, BiLSTM, and combination of Dense Layer for Gojek user's Review were successfully implemented in the data for this study. Data is classified using preprocessing, feature extraction, and Word2Vec embedding, which trans-form words into vector form for filtering. The accuracy test results demonstrated that our model gets the highest accuracy test results, with an accuracy of 87%, using BiLSTM with Word2Vec. After-wards, our model achieves the accuracy test results with an accuracy of 80% for LSTM with Word2Vec and an accuracy of 81% for BiLSTM + Dense Layer with Word2Vec.

This proves that the BiLSTM model with Word2Vec is able to have better performance than LSTM. Because BiLSTM incorporates information from both directions, it can therefore better capture complicated contextual linkages for tasks involving com-plex contextual dependencies, such as sentiment analysis. Furthermore, this can help understand context, especially for jobs that require awareness of the global context. The vanishing gradient or inflating gradient problem is a prevalent issue in the training of recurrent networks, such as LSTM. Because BiLSTM receives input from both directions, it allows for more efficient gradient propagation, which can help alleviate the vanishing gradient problem.

To examine models and algorithms that perform well when addressing particular sentiment analysis instances by using the data held to produce accurate results. Additional study is recommended, including case studies involving a variety of platforms, model comparisons, algorithms, other embedding models, as well as the combination of multiple models.

E. Acknowledgment

We express our gratitude and appreciate to Github assistance and Gojek in the form of datasets. This work was supported by Universitas Amikom Yogyakarta.

F. References

[1] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria dan U. R. Acharrya, “ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis,” *Elsevier - Future Generation Computer Systems,* 2021.

[2] J. Sangeetha dan U. Kumaran, “Using BiLSTM Structure with Cascaded Attention Fusion Model for Sentiment Analysis,” *Journal of Scientific & Industrial Research,* pp. 444-449, 2022.

[3] A. Onan, “Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks,” *Concurrency Computation Practice and Experience,* 2020.

[4] D. A. Nurdeni, I. Budi dan A. B. Santoso, “Sentiment Analysis on Covid19 Vaccines in Indonesia: From The Perspective of Sinovac and Pfizer,” dalam *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCI)*, Surabaya, Indonesia, 2021.

[5] V. A. Kharde dan S. S. Sonawane, “Sentiment Analysis of Twitter Data: A Survey of Techniques,” *International Journal of Computer Applications, 139(11),* 2016.

[6] A. Alsharef, K. Aggarwal, S. D. Koundal, H. Alyami dan D. Ameyed, “An Automated Toxicity Classification on Social Media Using LSTM and Word Embedding,” *Computational Intelligence and Neuroscience,* 2022.

[7] M. A. Riza dan N. Charibaldi, “Emotion Detection in Twitter Social Media Using Long Short Term Memory (LSTM) and Fast Text,” *International Journal of Artificial Intelligence & Robotics,* pp. 15-26, 2021.

[8] L. Xiaoyan, R. C. Raga dan S. Xuemei, “GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Review,” *Journal of Sensors,* pp. 1-12, 2022.

[9] D. D. N. Cahyo, F. Farasalsabila, V. B. Lestari, H. T. Lestari dan F. R. Al Islami, "Sentiment Analysis for IMDb Movie Review Using Support Vector Machine (SVM) Method," *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi, 8(2),* pp. 90-95, 2023.

[10] B. Subba dan S. Kumari, "A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings," *Computational Intelligence 38(2),* pp. 530-559, 2022.

[11] A. Rahman, E. Utami dan S. , "Sentimen Analisis Terhadap Aplikasi pada Google Playstore Menggunakan Algoritma Naive Bayes dan Algoritma Genetika," *Jurnal Komtika (Komputasi dan Informatika), 5(1),* pp. 60-71, 2021.

[12] D. R. Alghifari, M. Edi dan L. Firmansyah, "Implementasi Bidirectional LSTM untuk Analisis Sentimen," *Jurnal Manajemen Informatika (JAMIKA),* pp. 89-99, 2022.

[13] W. Aljedaani, F. Rustam, S. Ludi, A. Ouni dan M. W. Mkaouer, "Learning Sentiment Analysis for Accessibility User Reviews," dalam *2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*, 2021.

[14] C. Colón-Ruiz dan I. Segura-Bedmar, "Comparing deep learning architectures for sentiment analysis on drug reviews," *Journal of Biomedical Informatics 110,* 2020.

[15] N. M. Alharbi, N. S. Alghamdi, E. H. Alkhammash dan J. F. Al Amri, "Evaluation of Sentiment Analysis via Word Embedding and RNN Variants for Amazon Online Reviews," *Hindawi Mathematical Problems in Engineering,* 2021.

[16] L. Yang, Y. Li, J. Wang dan R. S. Siherratt, "Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning," *IEEE Access,* p. 23526, 2020.

[17] H. Zhao, Z. Liu, X. Yao dan Q. Yang, "A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach," *Information Processing and Management,* 2021.

[18] A. Iqbal, R. Amin, J. Iqbal, R. Alroobaea, A. Binmahfoudh dan M. Hussain, "Sentiment Analysis of Consumer Reviews Using Deep Learning," *Sustainability,* pp. 14(17), 10844, 2022.

[19] K. N. Gondhi, C. E. Sharma, A. H. Alharbi, R. Verma dan M. A. Shah, "Efficient Long Short-Term Memory-Based Sentiment Analysis of E-Commerce Reviews," *Hindawi - Computational Intelligence and Neuroscience,* 2022.

[20] R. N. Octaviani dan E. B. Setiawan, "The Effect of Sentiment on Bank Mandiri Stocks (BMRI) Using the Feature Expansion Method with Word2vec and Backpropagation Neural Network," dalam *2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS)*, Bandung, Indonesia, 2022.

[21] Ferselia, E. Utami dan A. Yaqin, "Sentiment Analysis of Shopee Food Application User Satisfaction Using the C4.5 Decision Tree Method," *Sinkron: jurnal dan penelitian teknik informatika,* vol. 8, no. 3, pp. 1554-1563, 2023.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall dan W. P. Kagelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research,* vol. 16, pp. 321-357, 2002.

[23] S. Poria, A. Hussain dan E. Cambria, Multimodal Sentiment Analysis, Socio-Affective Computing Vol. 8 Springer Nature, 2018.

[24] B. Wang, A. Wang, F. Chen, Y. Wang dan C. J. Kuo, "Evaluating Word Embedding Models: Methods and Experimental Results," *APSIPA transactions on signal and information processing,* p. 1 of 14, 2019.

[25] I. R. Hendrawan, E. Utami dan A. D. Hartanto, "Comparison of Word2vec and Doc2vec Methods for Text Classification of Product Reviews," dalam *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia, 2022.

[26] A. Nurdin, B. A. S. Aji, A. Bustamin dan Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," *Jurnal TEKNOKOMPAK, Vol. 14, No. 2,* pp. 74-79, 2020.

[27] S. Bhatia, M. Sharma dan K. K. Bhatia, "Sentiment Analysis and Mining of Opinions," *Internet of Things and Big Data Analytics Toward Next - Generation Intelligence. Studies in Big Data,* vol. 30, pp. 503-523, 2018.

[28] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, United Kingdom: Cambridge University Press, 2020.

[29] S. Siami-Namini, N. Tavakoli dan A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," dalam *2019 IEEE International Conference on Big Data (Big Data)*, 2019.

[30] A. Rostamian dan J. G. O'Hara, *Neural Computing and Applications,* vol. 34, no. 20, pp. 17193-17205, 2022.

[31] R. Rajendran dan B. S. Murugan, "An Efficient Pelican optimization based CNN-BiLSTM to Detect and Classify 3D Objects," dalam *2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)*, Chickballapur, India, 2022.

[32] G. Xu, Y. Meng, X. Qiu, Z. Yu dan X. Wu, "Sentiment Analysis of Comment Texts Based on BiLSTM," *IEEE Access,* 2019.

[33] S. H. Imam, S. Huhn, L. Hornuf dan R. Drechsler, "A Novel Default Risk Prediction and Feature Importance Analysis Technique for Marketplace Lending using Machine Learning," *Credit and Capital Markets–Kredit und Kapital,* vol. 56, no. 1, pp. 27-62, 2023.

[34] M. Schuster dan K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE TRANSACTIONS ON SIGNAL PROCESSING,* vol. 45, no. 11, pp. 2673-2681, 1997.

[35] J. Bhattacharjee, fastText Quick Start Guide: Get started with Facebook's library for text representation and classification., Birmingham: Packt Publishing Ltd., 2018.

[36] A. G. Yaman dan T. Can, "The effect of representative training dataset selection on the classification performance of the promoter sequences," dalam *Proceedings of the 6th International Symposium on Health Informatics and Bioinformatics*, Izmir, Turkey, 2011.

[37] G. SIhag, P. Yadav, V. Vijay, V. Delcroix, X. Siebert, S. K. Yadav dan F. Puisieux, "Advantages of Oversampling Techniques: A Case Study in Risk Factors for Fall Prediction," dalam *Information and Communication Technologies for Ageing Well and e-Health*, 2021.

[38] S. Wadhawan dan R. Maini, "ETCD: An effective machine learning based technique for cardiac disease prediction with optimal feature subset selection," *Knowledge-Based Systems,* vol. 255, p. 109709, 2022.