TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 23, No. 2, April 2025, pp. 416~425
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v23i2.26223

Comparison of word embedding features using deep learning in sentiment analysis

Jasmir[1], Errissya Rasywir[2], Herti Yani[3], Agus Nugroho[2]
[1]Departement of Computer Engineering, Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia
[2]Department of Informatic Engineering, Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia
[3]Department of Information System, Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia

Article Info

Article history:

Received Mar 24, 2024
Revised Dec 4, 2024
Accepted Jan 22, 2025

Keywords:

Deep learning
Sentiment analysis
Social media
Text classification
Word embedding

ABSTRACT

In this research, we use several deep learning methods with the word embedding feature to see their effect on increasing the evaluation value of classification performance from processing sentiment analysis data. The deep learning methods used are conditional random field (CRF), bidirectional long short term memory (BLSTM) and convolutional neural network (CNN). Our test uses social media data from Netflix application user comments. Through experimentation on different iterations of various deep learning techniques alongside multiple word embedding characteristics, the BLSTM algorithm achieved the most notable accuracy rate of 79.5% prior to integrating word embedding features. On the other hand, the highest accuracy value results when using the word embedding feature can be seen in the BLSTM algorithm which uses the word to vector (Word2Vec) feature with a value of 87.1%. Meanwhile, a very significant change in value increase was obtained from the FastText feature in the CNN algorithm. After all the evaluation processes were carried out, the best classification evaluation results were obtained, namely the BLSTM algorithm with stable values on all word embedding features.

Corresponding Author:

Jasmir
Department of Computer Engineering, Faculty of Computer Science, Universitas Dinamika Bangsa
St. Jendral Sudirman, Tehok, Jambi Selatan, Jambi, Indonesia
Email: ijay_jasmir@yahoo.com

1. INTRODUCTION

In recent decades, technological developments have experienced a rapid surge, especially since the emergence of the internet and personal computers in the 1980s. These technological advances have caused major changes in various sectors, including information and communication [1], [2]. The significant increase in internet technology has expanded the reach of information distribution. One aspect that supports this increase is social media, where users not only function as recipients of information but also as creators of information. The increase in the number of internet users in Indonesia is due to the various conveniences offered by social media and the internet. Through social media, people can access information and communicate very quickly.

The use of data from social media is the latest innovative step that provides an alternative data source outside of traditional data collection methods [3], [4]. Data collection via social media is considered to provide efficiency in many ways. This efficiency includes the costs that must be incurred for data acquisition, being able to obtain data in real time, and producing data that has more detailed information to describe the true opinion of the community [5]. Activities such as those above that are related to analyzing and responding to public opinion using data sourced from social media are called sentiment analysis [6], [7].

Sentiment analysis, which is a subset of natural language processing (NLP), uses machine learning methods to recognize and extract factual information from written text [8]. This analysis involves identifying emotional nuances and determining the overall sentiment—whether positive, neutral, or negative—expressed by the author. Applying sentiment analysis to larger data sets allows for a more comprehensive and in-depth level of analysis [9].

In NLP, computers do not have an innate understanding of textual language, so they need techniques to convert words into vectors for easier understanding. The process of representing word vectors remains an interesting area of research. This representation holds great significance as it profoundly influences the accuracy and efficacy of the constructed learning models. This word representation technique is included in the feature engineering section. Feature engineering in textual data has its own challenges due to the characteristics of unstructured text. The feature engineering strategy for textual data that is popularly used is known as the word embedding feature [10]–[12].

This word embedding feature is collaborated with several classification methods. There are many types of classifiers that are commonly used to classify sentiment analysis. The methods that are often used are machine learning methods [13], [14] and deep learning [15]. In this research, the types of methods used are deep learning methods, namely conditional random field (CRF) [16], bidirectional long short term memory (BLSTM) [17], and convolutional neural network (CNN) [18]. CRFs are used to build probabilistic models for sequential data segmentation and labeling. Because it is conditional, CRF is also used to ensure that inference is easy to do and also avoids the problem of label bias. BLSTM is used to find out the previous information process and find out the information process afterward. Meanwhile, CNN is used to see processing capabilities and evaluate classification performance on text data.

We evaluate the effectiveness of different classification methods by testing their performance using several types of word representations, namely word to vector (Word2Vec) [19], global vectors for word representation (GloVe) [20], and FastText [21]. The tests were conducted on a sentiment analysis dataset consisting of Netflix user comments. Netflix was chosen as the object of study due to its high popularity as a streaming platform, its large user base, and the variety of content it offers. This makes it a relevant topic for understanding user preferences for digital entertainment services. Analysis of user sentiment, both positive and negative, can provide valuable insights into their views on the quality of the service, interface, and content provided.

Similar studies that have been discussed include by Al-Smadi *et al*. [22] using several deep learning methods such as BLSTM-CRF combined with Word2Vec features and producing an F1-score of 66.32%. then BLSTM CRF combined with FastText features producing an F1-score of 69.98%. Then, Jang *et al.* [23] proposed a hybrid model of Bi-LSTM+CNN with Word2Vec, the test results showed that the proposed model produced more accurate classification results, as well as higher recall and F1 scores, than the multi-layer perceptron (MLP) model, CNN or individual LSTM and hybrid models. Furthermore, Iftikhar *et al.* [24] conducted experiments with several deep learning models combined with several word embedding features such as CNN+Glove, CNN+Word2Vec, LSTM+Glove, and LSTM+Word2Vec. The results of their research stated that the results of the combination of deep learning with the word embedding feature produced better performance. Based on the problems, we conducted research as well as the contribution of this research, namely to improve the evaluation value of the classification performance of deep learning methods, namely CRF, BLSTM, and CNN by using word embedding features, namely Word2Vec, GloVe, and FastText as techniques to improve the evaluation value of deep learning classification performance on machine learning datasets on social media data from Netflix application user comments.

2. MATERIAL AND METHOD

In order for this research to achieve maximum results, we have compiled a series of important steps that can produce the right model and not widen the direction in achieving the goal. The steps taken to obtain results that are in accordance with expectations are compiled in the form of a research framework. The research framework referred is presented in Figure 1.

2.1. Dataset

The dataset was obtained through a data collection process carried out by crawling. We utilize the Google Play Scraper Python library. To crawl data, the ID of the application from which data is to be retrieved is first required. In this case, Netflix has the ID 'com.netflix.mediaclient'. Furthermore, the selection of the language in the review is an important step, where this study only considers reviews in Indonesian. After selecting the language, the selection of reviews is based on the score. In this study, the reviews taken have a score range of 1 to 5. Furthermore, the order of reviews used is most relevant. The amount of data to be taken also needs to be determined. The data obtained has several attributes, including: reviewId, username, userImage, content, score, thumbsUpCount, reviewCreatedVersion, at, replyContent, answeredAt, and appVersion. However, not all of these attributes are required for this study. Therefore, irrelevant or unused attributes are removed to simplify the data. There are 4 attributes that will be used, namely username, score, date, and content. Figure 2 is a flow diagram of data collection. The data used is a raw dataset that will go through several pre-processing processes before becoming a dataset that is ready to use.

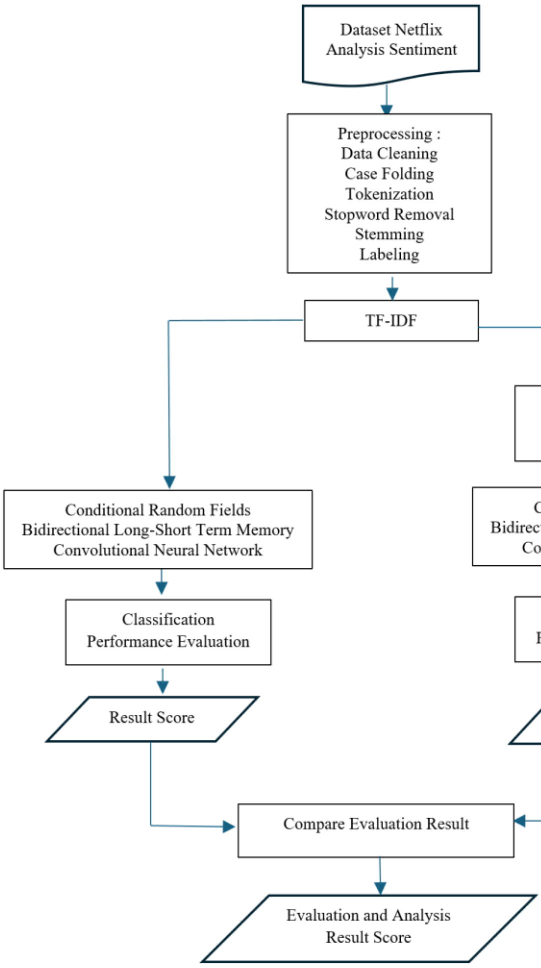

Figure 1. Research framework

2.2. Preprocessing

After getting the Netflix application user review data, the next step is to carry out the preprocessing stage before entering the sentiment classification stage. This process is important to ensure that the data used by sentiment classification models is clean, structured, and ready to use. The preprocessing stages carried out are data cleaning, case folding, tokenization, stopword removal, stemming, and labeling.

2.3. Word embedding

Every word is depicted as a numerical low-dimensional vector. Word embedding enables the capture of semantic details from extensive text corpora. These embeddings find application in diverse NLP tasks for optimal word representation. Notably, several algorithms exist for word embedding, including GloVe, Word2Vec, and FastText. For this research, we utilize pre-trained models encompassing all three features.

2.3.1. GloVe

GloVe is a co-occurrence matrix-based word representation learning technique that captures semantic relationships between words in a corpus. GloVe combines global statistics-based approaches (such as co-occurrence matrices) and local context-based methods (such as Word2Vec) to generate word embeddings in a vector space, allowing for more effective modeling of linear relationships between words [20], [25].

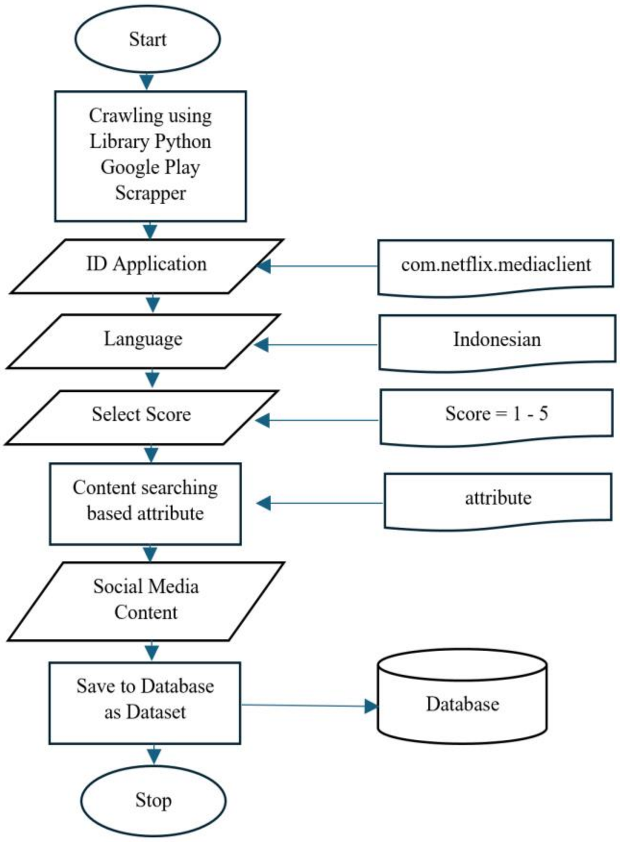

Figure 2. Data collection flow chart

2.3.2. Word2Vec

Word2Vec utilizes the occurrence of words in text to establish connections between them. For instance, it might associate words like “female” and “male” because they frequently occur in comparable contexts. Word2Vec operates through two architectural forms: context prediction, which forecasts the surrounding words based on a given word, and context-based prediction (Bag-of-words), which predicts words given a context. Essentially, Word2Vec takes a textual corpus as input and generates a word vector as output [19], [26].

2.3.3. FastText

Every word is depicted as a collection of n-gram characters, aiding in capturing the essence of shorter words and facilitating the embedding’s understanding of word prefixes and suffixes. Each n-gram character is linked with a vector representation, while words are depicted as the sum of these vector representations. FastText demonstrates strong performance, enabling rapid model training on extensive datasets and offering representations for words absent in the training data. In cases where a word is absent during model training, it can be decomposed into n-grams to acquire its embedding vector [21], [27].

2.4. Deep learning

2.4.1. Conditional random fields

CRFs belong to a class of discriminative models ideally suited for classification tasks wherein the current classification is impacted by contextual factors or adjacent states [28]. CRF finds application in named entity recognition [29], part-of-speech tagging, gene prediction, noise reduction, and object detection tasks. Discriminative models, also known as conditional models, are a subset of models commonly employed in statistical classification, particularly in supervised machine learning. Discriminative classifiers aim to model the observed data exclusively, learning classification from provided statistics. Approaches in supervised learning are typically classified into discriminative models or generative models. Discriminative models, in contrast to generative models, make fewer assumptions about distributions and place greater reliance on data quality [30], [31].

2.4.2. Bidirectional long short-term memory

Derived from the recurrent neural network (RNN), BLSTM [32] enhances the RNN architecture by introducing a "gateway" mechanism to regulate the flow of data. Primarily, the long short-term memory (LSTM) architecture comprises memory cells along with input gates, output gates, and forget gates. These elements are structured into a chain-like arrangement composed of RNN modules, which enables the smooth transfer of memory cells along the chain. Moreover, three separate gates are integrated to oversee and regulate the inclusion or inhibition of information into the memory cell [33].

2.4.3. Convolutional neural netrwork

The CNN is a form of regulated feed-forward neural network that autonomously learns feature engineering via the optimization of filters, also known as kernels. Unlike lower layer features, higher layer features are extracted from a broader context window. CNNs are sometimes called shift invariant or space invariant artificial neural networks (SIANN) because of their architecture, which involves convolution kernels or filters with shared weights moving across input features. This movement produces a feature map that is equivalent to translation. However, despite the terminology, many CNNs are not inherently translation invariant, mainly because of the downsampling operation applied to the input [34]–[36].

3. RESULTS AND DISCUSSION

This section summarizes the results of the experiments conducted according to the previously planned research flow. This experiment focuses on analyzing text data from social media using several deep learning methods combined with word embedding features. Training and testing data are divided with an 80:20 division scheme. This study tests deep learning methods with various variations of word embedding features. The deep learning method is applied as an approach to sentiment classification on text data. The types of deep learning methods used include CRF, BLSTM, and CNN. In addition, the word embedding features used include Word2Vec, GloVe, and FastText.

Table 1 explains the confusion matrix of CRF with three word embedding features and one without features. In CRF without features, the results are TP=406, FP=104, FN=91, and TN=299. This means that this model has a fairly low number of TP compared to the use of Word2Vec, GloVe, and FastText features. In addition, the FP value is quite high, indicating that the model tends to incorrectly identify negative data as positive. Then CRF with Word2Vec produces TP=512, FP=98, FN=69, and TN=221. The addition of the Word2Vec feature significantly increases TP (from 406 to 512), indicating that the model is better able to recognize positive data correctly. However, FP is still quite high (98), and the number of TN decreases compared to the model without features. This shows that Word2Vec improves the recognition of positive data but slightly decreases the ability to recognize negative data. In the CRF with GloVe section, there are results of TP 0=448, FP=91, FN=81, and TN=280. This indicates that GloVe provides more balanced results than Word2Vec. TP is lower than Word2Vec, but FP is also lower (91), indicating that the model is better at minimizing errors in classifying negative data. The number of TNs increases compared to Word2Vec. Then CRF with FastText which produces TP=473, FP=79, FN=59, and TN=289. It can be seen that FastText provides the best overall performance. TP and TN increase compared to GloVe, while FP and FN are the lowest among all models. This shows that FastText is very effective in improving the recognition of positive and negative data, with the least classification errors.

Table 1. Confusion matrix of CRF

Experiment	TP	FP	FN	TN
CRF without feature	406	104	91	299
CRF with Word2Vec	512	98	69	221
CRF with GloVe	448	91	81	280
CRF with FastText	473	79	59	289

Table 2 is a CRF test with 3 word embedding variants and one without word embedding. It can be seen that the accuracy without using features is lower than the model that uses features. This shows the importance of the embedding feature. In the CRF with Word2Vec section, there are the best results for Recall, which means that this model is able to capture more actual positive cases. Meanwhile, CRF with GloVe produces more stable performance in all metrics, although not the best. Then CRF with FastText gives the best value in accuracy, precision, and F1-Score. This model is the most optimal in producing correct predictions and maintaining a balance between precision and recall. These results show that the FastText feature provides a significant increase in accuracy (8.09%) and recall (8.84%), making it an excellent choice for improving the model's ability to capture true positives. Precision and F1-Score also increase quite significantly, supporting a balance between correct positive predictions and the ability to capture positive cases. Focus on the F1-Score metric, since F1-Score is a metric that combines precision and recall, it is very relevant for cases that require a balance between the two metrics, especially in classification tasks involving data with an imbalanced class distribution or cases where the balance between precision and recall is a priority. In these imbalanced datasets where one class is very dominant, accuracy may appear high because the model can ignore the minority class. F1-Score addresses this problem by taking the minority class into account. CRF with FastText has the highest F1-Score, indicating that this feature is optimal for producing a good balance.

Table 2. Comparison of CRF evaluation values with word embedding

Experiment	Accuracy	Precission	Recall	F1-Score
CRF without feature	78.33333333	79.60784	81.69014	80.63555
CRF with Word2Vec	81.44444444	83.93443	88.12392	85.97817
CRF with GloVe	80.88888889	83.11688	84.68809	83.89513
CRF with FastText	84.66666667	85.68841	88.90977	87.26937

Table 3 explains the confusion matrix of BLSTM with three word embedding features and one without features. In BLSTM without features, there are values of TP=481, FP=101, FN=83, and TN=235. This means that the model without features produces quite good performance, with a TP of 481. However, the FP is quite high (101), indicating that the model often misclassifies negative data as positive. In addition, the TN value is lower than the model using features, indicating a weaker ability to recognize negative data. In the BLSTM with Word2Vec section, there are values of TP=501, FP=97, FN=81, and TN=221. With the addition of the Word2Vec feature, the number of TP increases to 501, indicating that the model is better able to recognize positive data correctly than the model without features. However, the FP value is still quite high (97), meaning that the misclassification of negative data as positive remains quite significant. The decrease in the number of TN also indicates that negative data recognition is slightly impaired. Next, in BLSTM with GloVe, there are values of TP=480, FP=99, FN=79, and TN=242. The GloVe feature produces a slightly lower number of TP than Word2Vec (480 vs. 501), but the FN is also lower (79 vs. 81). In addition, the number of FP is smaller than Word2Vec (99 vs. 97), and TN increases to 242, indicating that the model is better at recognizing negative data than Word2Vec. Then BLSTM with FastText produces values of TP=505, FP=66, FN=50, and TN=279. The FastText model gives the best results among all methods. With the highest TP (505) and the lowest FN (50), this model is very effective in recognizing positive data. In addition, FP is the lowest (66), and TN is the highest (279), indicating that this model is also very good at recognizing negative data. This confirms that FastText improves overall performance.

Table 3. Confusion matrix of BLSTM

Experiment	TP	FP	FN	TN
BLSTM without feature	481	101	83	235
BLSTM with Word2Vec	501	97	81	221
BLSTM with GloVe	480	99	79	242
BLSTM with FastText	505	66	50	279

Table 4 is a summary table of the experimental results of the BLSTM method with three variants of word embedding features and one without features. In the BLSTM without feature section; this model serves as a baseline, and its performance is relatively good without additional features, but it can still be further improved by adding word embedding features. In the BLSTM with Word2Vec section, the model is slightly better than the baseline, with small improvements in accuracy, precision, recall, and F1-Score. The use of Word2Vec as an embedding feature improves the model's understanding of word relationships. In the BLSTM with GloVe section, the results are very similar to Word2Vec, but slightly lower in precision. This model shows better performance in terms of Recall, but not as good as the model with Word2Vec. Then in the BLSTM with FastText section: This is the best model, with significant improvements in all metrics. FastText provides clear improvements in precision, recall, and F1-Score, making it a very effective model in this classification.

The use of embedding features such as Word2Vec, GloVe, and FastText affects the improvement of model performance compared to the baseline model without features. The model with FastText shows the greatest improvement, especially in recall and F1-Score. BLSTM with FastText has the highest F1-Score (89.69%), indicating that this model is the best choice especially in the balance between accurate prediction and the model's ability to capture positive cases. The use of embedding features such as FastText can significantly improve performance compared to not using features or using other features such as Word2Vec and GloVe.

Table 4. Comparison of BLSTM evaluation values with word embedding

Experiment	Accuracy	Precission	Recall	F1-Score
BLSTM without feature	79.55555556	82.64605	85.28369	83.94415
BLSTM with Word2Vec	80.22222222	83.77926	86.08247	84.91525
BLSTM with GloVe	80.22222222	82.90155	85.86762	84.35852
BLSTM with FastText	87.11111111	88.44133	90.99099	89.69805

Table 5 explains the confusion matrix of CNN with three word embedding features and one without features. In CNN without features, there are values of TP=404, FP=128, FN=119, and TN=249. Without word embedding features, CNN produces the lowest performance. The number of TP is the lowest (404), while FP and FN are the highest (128 and 119). This shows that the model has many errors in recognizing both positive and negative data. In addition, the TN value is also quite low compared to the model with features. Next, the CNN with Word2Vec section has TP=512, FP=98, FN=70, and TN=220. The addition of Word2Vec significantly increases the number of TP to 512, indicating that the model is better able to recognize positive data than without features. However, the FP (98) and FN (70) values are still quite high, which means there is room for improvement in recognizing negative data. The decrease in the number of TNs compared to without features also shows that the model is slightly less effective in recognizing negative data. In the CNN with GloVe section, there are values of TP=446, FP=88, FN=87, and TN=279. This means that GloVe provides more balanced results than Word2Vec. FP decreases to 88, while TN increases significantly to 279, indicating a better ability to recognize negative data. However, TP is lower than Word2Vec (446 vs. 512), and FN is slightly higher than Word2Vec. Furthermore, CNN with FastText produces values of TP=463, FP=76, FN=74, and TN=287. FastText produces the best results among all methods. With high TP (463) and low FN (74), the model is very effective in recognizing positive data. In addition, FP is the lowest (76), and TN is the highest (287), indicating that this model is also very good at recognizing negative data.

Table 5. Confusion matrix of CNN

Experiment	TP	FP	FN	TN
CNN without feature	404	128	119	249
CNN with Word2Vec	512	98	70	220
CNN with GloVe	446	88	87	279
CNN with FastText	463	76	74	287

Table 6 is a summary table of the results of CNN experiments with three variants of word embedding features and one without features. This model shows the best improvement compared to the baseline model. FastText provides a very good balance between precision, recall, and F1-Score, with excellent results in all metrics. The use of embedding features such as Word2Vec, GloVe, and FastText significantly improves model performance compared to the baseline model that does not use additional features. The CNN with FastText model has the highest F1-Score (86.06%), which shows an optimal balance between prediction accuracy and the ability to capture positive cases, indicating that this model is very effective in balancing both aspects. The use of F1-Score in this case is because we want to maintain a balance between accuracy and precision and the model's ability to find all positive classes. The CNN with FastText model is the best choice for this model, with significant improvements in all metrics, especially in recall and precision. Thus, FastText provides better results than other embedding features such as Word2Vec and GloVe in optimizing text classification performance.

Table 6. Comparison of CNN evaluation values with word embedding

Experiment	Accuracy	Precission	Recall	F1-Score
CNN without feature	72.55555556	75.93985	77.24665	76.58768
CNN with Word2Vec	81.33333333	83.93443	87.97251	85.90604
CNN with GloVe	80.55555556	83.5206	83.6773	83.59888
CNN with FastText	83.33333333	85.89981	86.21974	86.05948

This study examines the impact of performance improvements, computationally BLSTM is very efficient, this is because the BLSTM process occurs sequentially and regularly, making it suitable for processing long texts and large datasets. With the word embedding feature, BLSTM can capture more interactions between features that may be ignored by CRF and CNN. While previous studies have investigated the impact of other features of the same method. the study did not explicitly discuss their effect on computational performance.

Based on the results of the three experiments, the BLSTM algorithm achieved the highest accuracy of 79.5%, while the CNN algorithm recorded the lowest accuracy of 72.5% before the word embedding feature was applied. After combining word embedding, BLSTM with the Word2Vec feature achieved the highest accuracy of 87.1%, while the lowest accuracy post-embedding was also seen in BLSTM using the GloVe and FastText features. By reviewing all classification evaluation metrics—accuracy, precision, recall, and F1 score—BLSTM emerged as the best performing algorithm, consistently producing stable results across all embeddings.

However, all tests still allow some false positives and false negatives, indicating potential areas for further research, such as minimizing these errors. Additional accuracy improvements can be achieved by tuning hyperparameters. An important observation is that, before embedding, CNN has the lowest performance, but after applying embedding, especially Word2Vec with BLSTM, the performance improves significantly. This may be due to the characteristics of CNN which are not well suited for text data, while BLSTM, which reads sequences bidirectionally, shows a high ability to process text in detail, resulting in superior performance.

4. CONCLUSION

Our study has highlighted the efficacy of pre-trained word embedding models in sentiment analysis. Through a series of experiments, we have demonstrated the ability of these models to achieve high levels of accuracy across diverse textual datasets. In our evaluation, various deep learning methods with different word embedding features were tested with CRF, BLSTM, and CNN algorithms. The use of word embedding features such as FastText, Word2Vec, and GloVe consistently improved the performance of various text classification models on CRF, BLSTM, and CNN compared to models without features. FastText was identified as the best feature based on the table results as it produced the most balanced classification with minimal error. FastText also produced highly accurate classification on both positive and negative data. Word2Vec excelled in recognizing positive data but tended to be less accurate on negative data. For limited computational resources, GloVe can be chosen as it provides balanced results with lower error compared to Word2Vec. GloVe offers a good balance with lighter computational requirements, suitable for reducing errors on negative data. The choice of word embedding features used can be tailored to the specific needs of the model and the classification objectives.

ACKNOWLEDGEMENTS

We would like to thank Yayasan Dinamika Bangsa Jambi for the moral and financial support in completing this research, and would like to thank the research and community service institution, Universitas Dinamika Bangsa Jambi for its facilities and annual work programs.

REFERENCES

[1] A. L. Guzman and S. C. Lewis, "Artificial intelligence and communication: A Human–Machine Communication research agenda," *New Media & Society*, vol. 22, no. 1, pp. 70–86, 2020, doi: 10.1177/1461444819858691.
[2] B. Jimada-Ojuolape and J. Teh, "Impact of the Integration of Information and Communication Technology on Power System Reliability: A Review," *IEEE Access*, vol. 8, pp. 24600–24615, 2020, doi: 10.1109/ACCESS.2020.2970598.
[3] R. Lozano-Blasco, M. Mira-Aladrén, and M. Gil-Lamata, "Social media influence on young people and children: Analysis on Instagram, Twitter and YouTube," *Comunicar*, vol. 30, no. 74, pp. 117–128, 2023, doi: 10.3916/C74-2023-10.
[4] B. T. K., C. S. R. Annavarapu, and A. Bablani, "Machine learning algorithms for social media analysis: A survey," *Computer Science Review*, vol. 40, p. 100395, 2021, doi: 10.1016/j.cosrev.2021.100395.
[5] S. M. Fernández-Miguélez, M. Díaz-Puche, J. A. Campos-Soria, and F. Galán-Valdivieso, "The impact of social media on restaurant corporations' financial performance," *Sustainability*, vol. 12, no. 4, pp. 1–14, 2020, doi: 10.3390/su12041646.
[6] H. R. Alhakiem and E. B. Setiawan, "Aspect-Bas1ed Sentiment Analysis on Twitter Using Logistic Regression with FastText Feature Expansion," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 5, pp. 840–846, 2022, doi: 10.29207/resti.v6i5.4429.
[7] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-Based Systems*, vol. 226, p. 107134, 2021, doi: 10.1016/j.knosys.2021.107134.
[8] A. Palanivinayagam, C. Z. El-Bayeh, and R. Damaševičius, "Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review," *Algorithms*, vol. 16, no. 5, pp. 1–28, 2023, doi: 10.3390/a16050236.
[9] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019, doi: 10.1109/ACCESS.2019.2909919.
[10] J. Jasmir, W. Riyadi, S. R. Agustini, Y. Arvita, D. Meisak, and L. Aryani, "Bidirectional Long Short-Term Memory and Word Embedding Feature for," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 505–510, 2022, doi: 10.29207/resti.v6i4.4005.
[11] S. Ruder, I. Vulić, and A. Søgaard, "A Survey of Cross-lingual Word Embedding Models," *Journal of Artificial Intelligence Research*, vol. 65, pp. 569–630, Aug. 2019, doi: 10.1613/jair.1.11640.
[12] Z. Zhuang, Z. Liang, Y. Rao, H. Xie, and F. L. Wang, "Out-of-vocabulary word embedding learning based on reading comprehension mechanism," *Natural Language Processing Journal*, vol. 5, no. August, p. 100038, 2023, doi: 10.1016/j.nlp.2023.100038.
[13] S. Rapacz, P. Chołda, and M. Natkaniec, "A method for fast selection of machine-learning classifiers for spam filtering,"

Electronics, vol. 10, no. 17, 2021, doi: 10.3390/electronics10172083.
[14] F. N. N. H. R. Passarella, S. Nurmaini, M. N. Rachmatullah, and H. Veny, "Development of a machine learning model for predicting abnormalities of commercial airplanes," *Data Science and Management*, p. 100137, 2023, doi: 10.1016/j.jsamd.2023.100613.
[15] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, 2019, doi: 10.1016/j.neunet.2018.12.002.
[16] J. Jasmir, S. Nurmaini, R. F. Malik, and B. Tutuko, "Bigram feature extraction and conditional random fields model to improve text classification clinical trial document," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 19, no. 3, pp. 886–892, 2021, doi: 10.12928/TELKOMNIKA.v19i3.18357.
[17] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classificatio," *Neurocomputing*, 2019, doi: 10.1016/j.neucom.2019.01.078.
[18] M. Akbar, S. Nurmaini, and R. U. Partan, "The deep convolutional networks for the classification of multi-class arrhythmia," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 2, pp. 1325–1333, 2024, doi: 10.11591/eei.v13i2.6102.
[19] R. Rahmanda and E. B. Setiawan, "Word2Vec on Sentiment Analysis with Synthetic Minority Oversampling Technique and Boosting Algorithm," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 599–605, 2022, doi: 10.29207/resti.v6i4.4186.
[20] A. George, H. B. B. Ganesh, M. A. Kumar, and K. P. Soman, *Significance of global vectors representation in protein sequences analysis*, Springer International Publishing, vol. 31, 2019, doi: 10.1007/978-3-030-04061-1_27.
[21] I. N. Khasanah, "Sentiment Classification Using fastText Embedding and Deep Learning Model," *Procedia CIRP*, vol. 189, pp. 343–350, 2021, doi: 10.1016/j.procs.2021.05.103.
[22] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, and Y. Jararweh, "Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 8, pp. 2163–2175, 2019, doi: 10.1007/s13042-018-0799-4.
[23] B. Jang, M. Kim, G. Harerimana, S. U. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining word2vec CNN and attention mechanism," *Applied Sciences*, vol. 10, no. 17, p. 5841, 2020, doi: 10.3390/app10175841.
[24] S. Iftikhar, B. Alluhaybi, M. Suliman, A. Saeed, and K. Fatima, "Amazon products reviews classification based on machine learning, deep learning methods and BERT," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 21, no. 5, pp. 1084–1101, 2023, doi: 10.12928/TELKOMNIKA.v21i5.24046.
[25] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection," *Procedia Computer Science*, vol. 207, pp. 769–778, 2022, doi: 10.1016/j.procs.2022.09.132.
[26] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2vec model analysis for semantic similarities in English words," *Procedia Computer Science*, vol. 157, pp. 160–167, 2019, doi: 10.1016/j.procs.2019.08.153.
[27] M. A. Raihan and E. B. Setiawan, "Aspect Based Sentiment Analysis with FastText Feature Expansion and Support Vector Machine Method on Twitter," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 591–598, 2022, doi: 10.29207/resti.v6i4.4187.
[28] Q. Zhang, Y. Cao, and H. Yu, "Parsing citations in biomedical articles using conditional random fields," *Computers in biology and medicine*, vol. 41, no. 4, pp. 190–194, 2011, doi: 10.1016/j.compbiomed.2011.02.005.
[29] W. Lee, K. Kim, E. Y. Lee, and J. Choi, "Conditional random fields for clinical named entity recognition: A comparative study using Korean clinical texts," *Computers in Biology and Medicine*, vol. 10, pp. 7–14, 2018, doi: 10.1016/j.compbiomed.2018.07.019.
[30] P. Corcoran, P. Mooney, and M. Bertolotto, "Linear street extraction using a Conditional Random Field model," *Spatial Statisticsvol*, vol. 14, pp. 532–545, 2015, doi: 10.1016/j.spasta.2015.10.003.
[31] C. Jiang, M. Maddela, W. Lan, Y. Zhong, and W. Xu, "Neural CRF model for sentence alignment in text simplification," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7943–7960, doi: 10.18653/v1/2020.acl-main.709.
[32] K. S. Tai, R. Socher, and C. D. Manning, "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks," *arXiv preprint*, 2015, doi: 10.48550/arXiv.1503.00075.
[33] Z. Dai, X. Wang, P. Ni, Y. Li, G. Li, and X. Bai, "Named Entity Recognition Using BERT BiLSTM CRF for Chinese Electronic Health Records," *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Suzhou, China, 2019, pp. 1–5, doi: 10.1109/CISP-BMEI48845.2019.8965823.
[34] D. T. Putra and E. B. Setiawan, "Sentiment Analysis on Social Media with Glove Using Combination CNN and RoBERTa," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 3, pp. 457–563, 2023, doi: 10.29207/resti.v7i3.4892.
[35] J. Zhang, F. Liu, W. Xu, and H. Yu, "Feature fusion text classification model combining CNN and BiGRU with multi-attention mechanism," *Future Internet*, vol. 11, no. 11, 2019, doi: 10.3390/fi11110237.
[36] J. Yao, C. Wang, C. Hu, and X. Huang, "Chinese Spam Detection Using a Hybrid BiGRU-CNN Network with Joint Textual and Phonetic Embedding," *Electronics*, vol. 11, no. 15, 2022, doi: 10.3390/electronics11152418.

BIOGRAPHIES OF AUTHORS

Jasmir is senior lecture at Universitas Dinamika Bangsa Jambi, Indonesia. He received his Bachelor in Computer Engineering in 1995 and Master degree in Information Technology in 2006 from Universitas Putra Indonesia YPTK Padang, Indonesia. He receives a Doctor in Informatics Engineering at Universitas Sriwijaya Palembang, Indonesia in 2022. His research interest is data mining, machine learning and deep learning for natural language processing, and its application. He can be contacted at email: ijay_jasmir@yahoo.com.

Errissya Rasywir received the Bachelor degree (S.Kom) in Computer Science from the Sriwijaya University. She received the Master degree (M.T) in Informatics Master STEI from the Institut Teknologi Bandung (ITB). She is a lecture of computer science in the Informatics Engineering, Dinamika Bangsa University (UNAMA). She is currently studying for a Doctorate in Computer Science at Sriwijaya University. In addition, she is serving as head of the research group (LPPM) on UNAMA. Her research interests are in data mining, artificial intelligent (AI), natural languange proccessing (NLP), machine learning, and deep learning. She can be contacted at email: errissya.rasywir@gmail.com.

Herti Yani is a lecture at Universitas Dinamika Bangsa Jambi, Indonesia. She received his Bachelor in Information System in Universitas Dinamika Bangsa Jambi in 2009 and Master degree in Magister System Information in Universitas Dinamika Bangsa Jambi, Indonesia in 2011. She is currently studying for a Doctorate in Computer Science at Satya Wacana Christian University. Her research interest are in database, artificial intelligence, and machine learning. She can be contacted at email: adeherti@unama.ac.id.

Agus Nugroho is lecture at Universitas Dinamika Bangsa Jambi, Indonesia. He received his Bachelor in Informatics Engineering in Universitas Dinamika Bangsa Jambi in 2011 and Master degree in Magister of Informatics Engineering in STMIK AMIKOM Yogyakarta, Indonesia in 2013. His research interest are in multimedia, artificial intelligence, and machine learning. She can be contacted at email: agusnugroho0888@gmail.com.