Word Embedding Feature for Improvement Machine Learning Performance in Sentiment Analysis Disney Plus Hotstar Comments

Jasmir, Nurhadi, Eni Rohaini, M Riza Pahlevi, Daniel Sintong Pardamean Simanjuntak
Departement of Computer Engineering, Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia

ARTICLE INFO

Article history:

Received May 09, 2024
Revised June 25, 2024
Published July 02, 2024

Keywords:

Text Classification;
Machine Learning Evaluation;
Word Embedding;
Sentiment Analysis;
Social Media Analysis

ABSTRACT

In this research we apply several machine learning methods and word embedding features to process social media data, specifically comments on the Disney Plus Hotstar application. The word embedding features used include Word2Vec, GloVe, and FastText. Our aim is to evaluate the impact of these features on the classification performance of machine learning methods such as Naive Bayes (NB), K-Nearest Neighbor (KNN), and Random Forest (RF). NB is very simple and efficient and very sensitive to feature selection. Meanwhile, KNN is known for its weaknesses such as biased k values, overly complex computations, memory limitations, and ignoring irrelevant attributes. Then RF has a weakness, namely that the evaluation value can change significantly with just a slight change in the data. Feature selection in text classification is crucial for enhancing scalability, efficiency, and accuracy. Our testing results indicate that KNN achieved the highest accuracy both before and after feature selection. The FastText feature led to the highest performance for KNN, yielding balanced accuracy, precision, recall, and F1-score values.

Corresponding Author:

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

1. INTRODUCTION

The integration of data derived from social media represents a significant advancement, offering an alternative data source to traditional data collection methods [1], [2], [3]. Social media data collection is efficient in various aspects, including cost-effectiveness, real-time data acquisition, and the ability to capture detailed community opinions [4], [5]. The analysis of public responses and opinions using social media data is known as sentiment analysis [6], [7], [8].

Sentiment analysis, a subset of natural language processing (NLP), employs machine learning methods to identify and extract factual details and emotional nuances from written text, determining the general sentiment—positive, neutral, or negative—expressed by the writer [9], [10], [11]. Applying sentiment analysis to extensive textual datasets, such as social media updates or user comments, allows for comprehensive analysis of public sentiment [12], [13], [14].

Several previous studies have explored sentiment analysis. For instance, Elik Hari Muktafin analyzed public service customer satisfaction using KNN with the TF-IDF feature, achieving an accuracy of 74% [15]. Heru Agus Santoso *et al.* examined sentiment analysis of hoax news using Naive Bayes, resulting in an accuracy of 77% [9]. Meanwhile, M. Ali Fauzi conducted sentiment analysis in Indonesian using Random Forest, achieving an average Out of Bag (OOB) value of 82,9% [16]. Ari Basuki conducted research on Sentiment Analysis of Customer Reviews of Delivery Service Providers on Twitter Using Naive Bayes

Classification and produced a low accuracy of 50.6% [17]. Kartikasari Kusuma Agustiningsih analyzed Indonesian public sentiment towards the COVID-19 vaccine on Twitter using BiLSTM and word embedding features, namely FastText and Glove. The combination of BLSTM and FastText produces an accuracy of 75.76%. The combination of BLSTM and GLove produces an accuracy of 74.70% [17]. These studies indicate potential for improving classification performance through experiments with various machine learning methods and features.

In NLP, computers lack an inherent understanding of textual language, necessitating techniques to convert words into vectors for effective processing. Word vector representation remains a compelling area of ongoing research, as it significantly impacts the accuracy and efficacy of learning models. This technique is a crucial aspect of feature engineering, which is particularly challenging in the context of unstructured text. A popular strategy in this domain is the use of word embedding features [18], [19], [20].

This research integrates word embedding features with several classification methods. Common classifiers for sentiment analysis include machine learning methods [21], [22], [23] and deep learning method [24], [25], [26]. In this study, we focus on machine learning methods, specifically Naive Bayes, KNN, and Random Forest. However, each method has its drawbacks. Naive Bayes struggles with complex dimensions, leading to lower classification accuracy and biased results [27]. K-Nearest Neighbor is highly dependent on feature scaling [28], [29]. Random Forest requires substantial computing resources for high accuracy, leading to longer prediction times [30], [31].

We evaluated the efficacy of various classifier methods by testing their performance with different word embedding features: Word2Vec [32], GloVe [33], and FastText [34]. The experimental process utilized a sentiment analysis dataset from Netflix user comments. Netflix was chosen due to its dominant popularity in streaming services, large user base, and diverse content, making it a relevant subject for understanding user preferences in digital entertainment. Analyzing user sentiment—whether positive or negative—provides valuable insights into their perceptions of the service, interface, and content.

The contribution of this research is that we conducted sentiment analysis of comments on the Disney Plus Hotstar application. Apart from that, the contribution of this research is increasing the evaluation value of the classification performance of machine learning methods with word embedding features and analyzing their performance.

2. METHODS

To achieve optimal results, we followed a series of essential steps to develop an appropriate model and maintain a clear focus on our research objectives. The steps involved in the classification process are illustrated in Fig. 1.

- Initially, data collection required the use the data for this research was collected from user comments on the Disney Plus Hotstar application on the Google Play Store
- Next, the text undergoes preprocessing, an important step in preparing it for training and testing, which involves Data Cleaning, Case Folding, Tokenization, Stopword Removal, Stemming and Labeling.
- After preprocessing, the research moves on to conducting training and testing, which involves two approaches: one without utilizing features and the other using features. In the feature-based approach, each word embedding feature is applied with each classifier.. The analysis considers parameters such as accuracy, precision, recall and F1-score. Testing is carried out using the results of each feature in each classifier.
- After implementing all the methodologies, the next step is to compare the results of the training and testing process for each approach. Each classifier integrates each word embedding feature
- The process ends with an evaluation of the training and testing procedures, as well as an analysis of the resulting classification performance.

2.1. Dataset

The data for this research was collected from user comments on the Disney Plus Hotstar application on the Google Play Store using Python and web scraping techniques. We used the Google Play Scraper Python library for data crawling, as depicted in Fig. 2. The attributes used in this study include:

- Username: The name of the user who posted the comment.
- Score: The rating given by the user to the application, ranging from 1 to 5.
- Content: The text of the user comments regarding the application.
- The raw data underwent several preprocessing steps to create a ready-to-use dataset.

2.2. Preprocessing

After collecting user review data, we performed preprocessing to ensure the data was clean, structured, and ready for sentiment classification. The preprocessing stages included:

- Data Cleaning: Removing noise and irrelevant information.
- Case Folding: Converting all text to lowercase.
- Tokenization: Splitting text into individual words or tokens.
- Stopword Removal: Removing common words that do not contribute to sentiment (e.g., "and," "the").
- Stemming: Reducing words to their base or root form.
- Labeling: Assigning sentiment labels to the comments.

2.3. Word Embedding

Each word was represented as a low-dimensional numerical vector, capturing semantic details from extensive text corpora. We used pre-trained models for three word embedding techniques:

2.3.1. GloVe

GloVe (Global Vectors for Word Representation) relies on co-occurrence and matrix factorization to produce vectors, establishing statistical relationships between words by constructing a large matrix of word co-occurrences [33], [35].

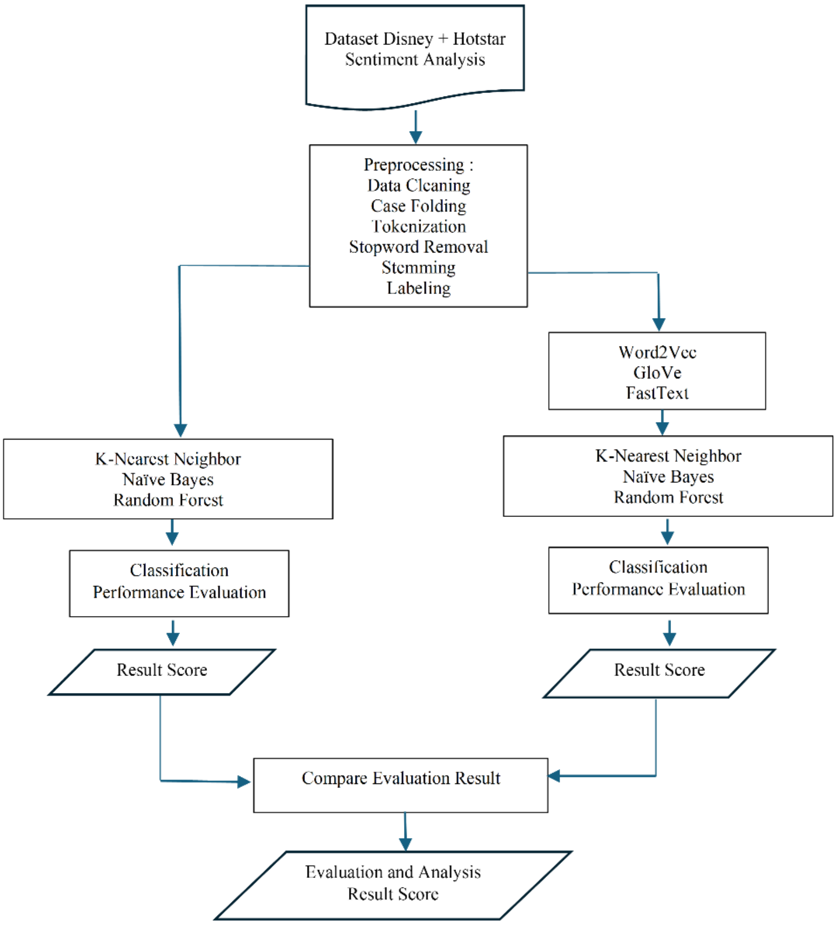
Dataset Disney + Hotstar Sentiment Analysis
Preprocessing : Data Cleaning Case Folding Tokenization Stopword Removal Stemming Labeling
Word2Vec GloVe FastText
K-Nearest Neighbor Naïve Bayes Random Forest
K-Nearest Neighbor Naïve Bayes Random Forest
Classification Performance Evaluation
Classification Performance Evaluation
Result Score
Result Score
Compare Evaluation Result
Evaluation and Analysis Result Score

Fig. 1. Research Framework

2.3.2. Word2Vec

Word2Vec creates vectors based on word co-occurrences, using either context prediction (predicting surrounding words from a given word) or the Bag-of-Words model (predicting words from a given context) [36], [32].

2.3.3. FastText

FastText represents each word as a collection of n-gram characters, capturing the essence of shorter words and understanding prefixes and suffixes. This approach allows FastText to handle words not present in the training data by decomposing them into n-grams [34], [37].

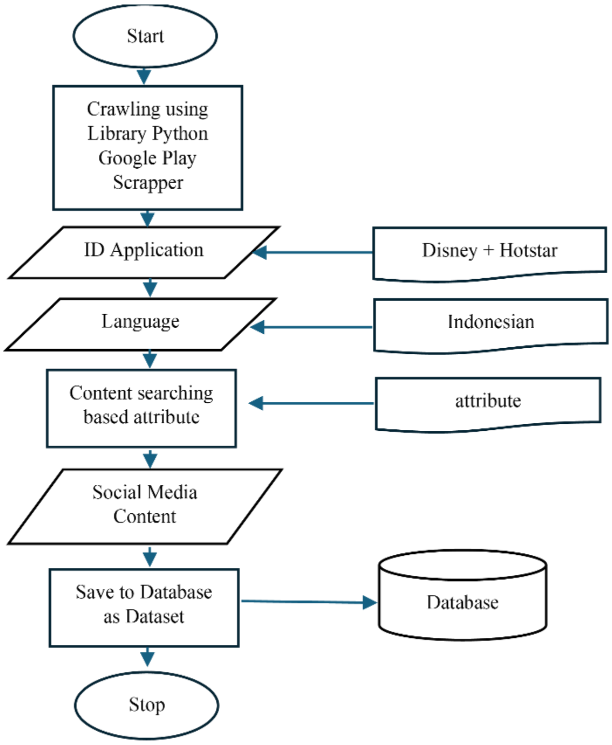
Start
Crawling using Library Python Google Play Scrapper
ID Application
Disney + Hotstar
Language
Indonesian
Content searching based attribute
attribute
Social Media Content
Save to Database as Dataset
Database
Stop

Fig. 2. Data collection flow chart

2.4. Learning Model

We evaluated the performance of three machine learning algorithms using the word embedding features:

2.4.1. K-Nearest Neighbor

KNN classifies objects based on the proximity of training data points [38], [39], [40]. It does not involve an offline training phase [41] . Instead, it stores all training documents and computes distances during the prediction phase [42]. KNN assigns classes based on the closest neighbors and their categories [43].

2.4.2. Naïve Bayes

Naïve Bayes is a simple probabilistic classifier that assumes feature independence [44], [45]. It balances performance with computational efficiency and performs well with small sample sizes due to inherent regularization [46], [47]. However, it struggles with interactions between features [48].

2.4.3. Random Forest

Random Forest is an ensemble learning method that constructs multiple independent decision trees [49], [50]. Each tree votes on the class of a test example, and the majority vote determines the final prediction [51]. Random Forests address overfitting by aggregating diverse trees created through random feature and data selection [52].

3. RESULTS AND DISCUSSION

This section provides an overview of the outcomes and deliberations stemming from experiments conducted according to the research framework outlined in the preceding section. The experiments revolve around the assessment of social media text data using a variety of machine learning methods and word embedding features. With 80:20 split validation. The tests carried out in this research included machine learning testing with variations of word embeddings. Machine Learning is a sentiment classification method for text data used in this research. The following types of machine learning methods are used, namely: Naive

Bayes (NB), K-Nearest Neighbor (KNN) and Random Forest (RF). For word embedding, we use 3 features, namely: Word2Vec, GloVe and Fast Text.

Table 1 is the test results of sentiment analysis of comments from users of the Disney Plus Hotstar Application using the Naive Bayes algorithm without using features. The test results are stored in a confusion matrix with each evaluation result. You can see that the results are false positive = 111 and false negative = 95. These values are considered very high so they result in low accuracy values.

Table 1. Confusion Matrix of NB

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 351	FP = 111
	Class = No	FN = 95	TN = 343

Table 2 is the test results of sentiment analysis of comments from users of the Disney Plus Hotstar Application using the Naive Bayes algorithm and using the word2vec feature. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 99 and false negative = 71. These values are in the ideal area for increasing the classification performance evaluation value so that it has an impact on higher accuracy values.

Table 2. Confusion Matrix of NB after using the Word2Vec feature

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 509	FP = 99
	Class = No	FN = 71	TN = 221

Table 3 is the test result of sentiment analysis of comments from users of the Disney Plus Hotstar Application using the Naive Bayes algorithm and using the Glove feature. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 92 and false negative = 82. These values are also in the ideal area for increasing the evaluation value of classification performance.

Table 3. Confusion Matrix of NB after using GloVe feature

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 445	FP = 92
	Class = No	FN = 82	TN = 281

Table 4 is the test result of sentiment analysis of comments from users of the Disney Plus Hotstar Application using the Naive Bayes algorithm and using the FastText feature. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 81 and false negative = 71. This value is very ideal for calculating the increase in classification performance evaluation value and produces the best value for Naive Bayes.

Table 4. Confusion Matrix of NB after using FastText feature

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 471	FP = 81
	Class = No	FN = 71	TN = 287

Fig. 3 illustrates the experimental results regarding sentiment analysis of Disney Plus Hotstar Application user data, which consists of 900 records. The analysis was performed using Naive Bayes techniques with three different word embedding features, in addition to a configuration without any features. In this experiment, it was seen that there was an increase in the value before using the feature and after using the word embedding feature. The highest word embeddings are generated by the FastText feature. All features are able to increase the classification performance evaluation value and generally produce values that tend to be stable, namely the Glove feature and the FastText feature, only the word2vec feature is unstable.

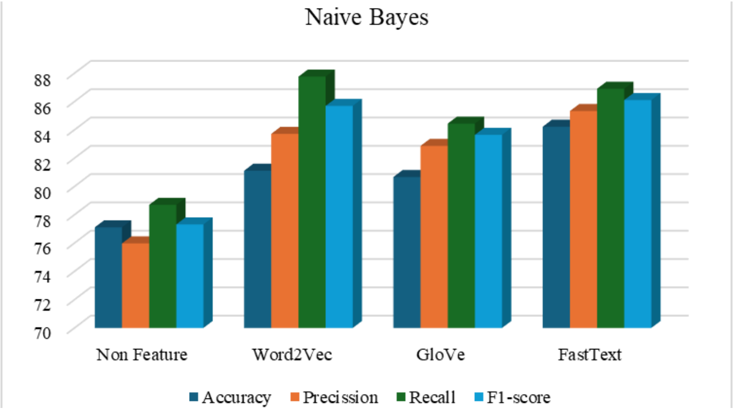

Fig. 3. Comparison graph of NB evaluation values with word embedding

In general it can be stated that Naive Bayes often works well on text data because its conditional independence assumption fits well with word representation models (Bag-of-Words or TF-IDF). However, when using the Word Embeddding feature, Naive Bayes may not fully utilize this information.

Next Table 6 is the result of testing sentiment analysis of comments from users of the Disney Plus Hotstar Application using the KNN algorithm without using features. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 103 and false negative = 85. This value is considered very high, resulting in a low accuracy value.

Table 6. Confusion Matrix of KNN

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 471	FP = 103
	Class = No	FN = 85	TN = 241

Table 7 is the result of testing sentiment analysis of comments from users of the Disney Plus Hotstar Application using the KNN algorithm and using the word2vec feature. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 95 and false negative = 79. This value is enough to increase the classification performance evaluation value, but it is not significant.

Table 7. Confusion Matrix of KNN after using Word2Vec feature

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 507	FP = 95
	Class = No	FN = 79	TN = 219

Table 8 is the result of testing sentiment analysis of comments from users of the Disney Plus Hotstar Application using the KNN algorithm and using the Glove feature. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 97 and false negative = 77. These values are also sufficient to increase the classification performance evaluation value, but are also not significant.

Table 8. Confusion Matrix of KNN after using GloVe feature

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 489	FP = 97
	Class = No	FN = 77	TN = 237

Table 9 is the result of testing sentiment analysis of comments from users of the Disney Plus Hotstar Application using the Naive Bayes algorithm and using the FastText feature. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 64 and false negative = 48. These values are ideal for calculating increased classification performance evaluation values and producing the best value for KNN.

Table 9. Confusion Matrix of KNN after using FastText feature

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 511	FP = 64
	Class = No	FN = 48	TN = 277

Fig. 4 explains the experimental results of Disney Plus Hotstar Application user sentiment analysis data of 900 records, using the KNN method with three word embedding features and one without using features. In this experiment, it can be seen that there was also an increase in the value before using the feature and after using the word embedding feature. The highest word embedding is also produced by the FastText feature. All word embedding features are able to increase the evaluation value of KNN, and generally produce stable values.

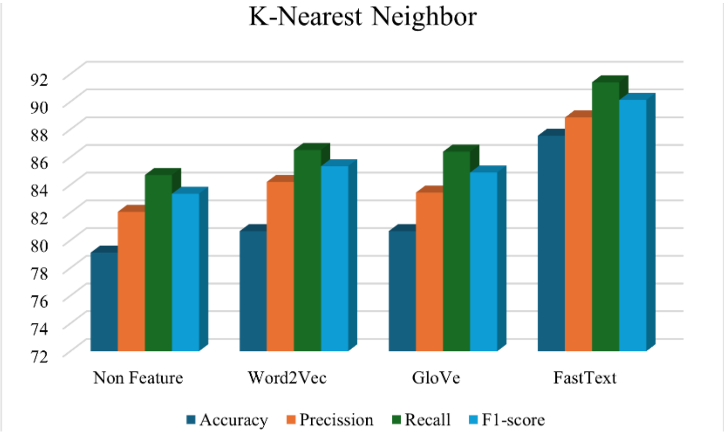

Fig. 4. Comparison graph of KNN evaluation values with word embedding

In general it can be stated that KNN works by finding the shortest distance between feature vectors. With the Word Embedding feature, KNN can provide good results if the distance between vectors effectively separates the classes. However, KNN can be slow and less efficient on large data because it has to calculate the distance to all points in the training dataset

The following is Table 11 of the results of testing sentiment analysis of user comments. The Disney Plus Hotstar Application uses the RF algorithm without using features. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the false positive results = 149 and false negative = 154. This value is very high so it produces a very low accuracy value

Table 11. Confusion Matrix of RF

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 355	FP = 149
	Class = No	FN = 154	TN = 242

Table 12 is the result of testing sentiment analysis of comments from users of the Disney Plus Hotstar Application using the RF algorithm and using the Word2Vec feature. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 76 and false negative = 74. These values are very good for getting an increase in the classification performance evaluation value and producing the best value for RF.

Table 12. Confusion Matrix of RF after using Word2Vec feature

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 463	FP = 76
	Class = No	FN = 74	TN = 287

Table 13 is the result of testing sentiment analysis of comments from users of the Disney Plus Hotstar Application using the RF algorithm and using the Glove feature. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 113 and false negative =

95. Even though there is a rather significant increase compared to without using the feature, this value is still considered and produces an accuracy value that is not yet good.

Table 13. Confusion Matrix of RF after using GloVe feature

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 404	FP = 113
	Class = No	FN = 95	TN = 288

Table 14 is the result of testing sentiment analysis of comments from users of the Disney Plus Hotstar Application using the RF algorithm and using the FastText feature. The test results are stored in a confusion matrix with each evaluation result. It can be seen that the results are false positive = 141 and false negative = 122. This value is still considered not good, because it produces a value that is still low.

Table 14. Confusion Matrix of RF after using FastText feature

		Actual Class	
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP = 413	FP = 141
	Class = No	FN = 122	TN = 224

Fig. 5 explains the experimental results of Disney Plus Hotstar Application user sentiment analysis data of 900 records, using the Random Forest algorithm with three word embedding features and one without using features. In this experiment, it can be seen that there was also an increase in the value before using the feature and after using the word embedding feature. In this case, the highest word embedding is produced by the Word2Vec feature. All word embedding features are able to increase the evaluation value of Random Forest, and generally produce stable values.

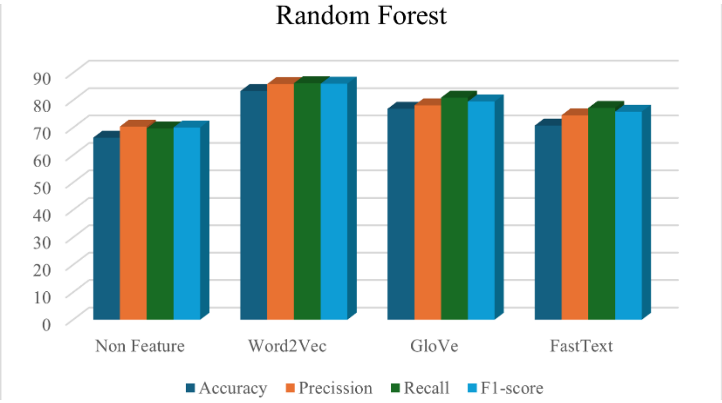

Fig. 5. Comparison graph of RF evaluation values with word embedding

In general it can be said that Random Forest tends to provide better performance because it utilizes a large number of decision trees and random features to reduce overfitting. With the word embedding feature, Random Forest can capture more interactions between features that Naive Bayes or KNN might ignore.

From all the machine learning tests carried out, it can be seen that the KNN algorithm achieved the highest accuracy level of 79.11%, while the Random Forest algorithm produced the lowest accuracy of 66.33%, before including any word embedding features. Meanwhile, the highest accuracy results after using the word embedding feature were obtained from the KNN algorithm with the FastText feature with a value of 87.55% and the lowest accuracy was obtained from the Random Forest algorithm with the FastText feature. After looking at all classification performance evaluation values, namely accuracy, precision, recall and f1-score, the best algorithm is the KNN algorithm with stable evaluation results on all word embeddings. All tests still tolerate false positive and false negative errors. All algorithms still use the original parameters. This could be a gap for further research such as reducing the value of false positives or false negatives. The gap to improve accuracy can also be achieved by tuning all hyperparameters.

Furthermore, the best word embedding produced in this experiment was FastText which gave the highest score on the KNN algorithm. This is very much in line with how each method works. One of fastText's distinctive features lies in its incredible speed and efficiency. The design facilitates fast training on large corpora, making it ideal for real-time applications and large-scale datasets. Additionally, the ability to generate embeddings for subword units significantly increases its usefulness, especially in scenarios involving rare or invisible words. These attributes have proven invaluable across a wide range of linguistic landscapes and specific domains..

The experiments included machine learning testing with variations of word embeddings. The machine learning methods used for sentiment classification in this research are Naive Bayes (NB), K-Nearest Neighbor (KNN), and Random Forest (RF). For word embedding, we utilized three features: Word2Vec, GloVe, and FastText. The results demonstrate that incorporating word embedding features significantly improves the performance of sentiment classification models. FastText consistently provided the best results across different algorithms due to its ability to capture subword information and its efficiency in handling large datasets.

Despite the improvements, all models still exhibited false positives and false negatives. Further research could focus on reducing these errors by tuning hyperparameters and exploring advanced preprocessing techniques. Additionally, enhancing the training datasets with more diverse samples may further improve the model's robustness. The findings confirm that FastText's speed and subword representation capability make it particularly effective for sentiment analysis tasks, aligning well with the results observed in this study.

4. CONCLUSION

In this paper, we investigated the impact of various machine learning methods combined with several word embedding features on the classification performance of sentiment analysis for Disney Plus Hotstar user comments. Specifically, we compared the performance of Naive Bayes (NB), K-Nearest Neighbor (KNN), and Random Forest (RF) algorithms before and after incorporating word embedding features such as Word2Vec, GloVe, and FastText. The experiments demonstrated that all machine learning methods experienced an improvement in classification performance metrics—accuracy, precision, recall, and F1-score—when word embedding features were applied. Among the tested methods, KNN achieved the highest accuracy of 79.11% without any word embedding features. After incorporating word embeddings, KNN with the FastText feature yielded the highest accuracy of 87.55%. Additionally, KNN exhibited balanced performance across all evaluation metrics (accuracy, precision, recall, and F1-score) when combined with the FastText word embedding feature, underscoring its robustness and effectiveness in sentiment analysis tasks. These findings highlight the significant role of word embedding features in enhancing the performance of machine learning algorithms for sentiment classification. FastText, in particular, proved to be the most effective word embedding feature, likely due to its ability to capture subword information and handle large datasets efficiently. Future work could focus on further reducing false positive and false negative rates by fine-tuning hyperparameters and employing more advanced preprocessing techniques. Expanding the dataset with more diverse samples may also enhance the model's robustness and generalizability.

REFERENCES

[1] R. L. -Blasco, M. M. -Aladrén., and M. G. -Lamata, "Social media influence on young people and children: Analysis on Instagram, Twitter and YouTube," *Comunicar*, vol. 30, no. 74, pp. 117–128, 2023, http://doi.org/10.3916/C74-2023-10.

[2] T. Mantoro, M. A. Permana, and M. A. Ayu, "Crime index based on text mining on social media using multi classifier neural-net algorithm," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 20, no. 3, pp. 570–579, 2022, https://doi.org/10.12928/TELKOMNIKA.v20i3.23321.

[3] S. Assegaff, E. Rasywir, and Y. Pratama, "Experimental of vectorizer and classifier for scrapped social media data," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 21, no. 4, pp. 815–824, 2023, https://doi.org/10.12928/TELKOMNIKA.v21i4.24180.

[4] 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," *Sustain.*, vol. 12, no. 4, pp. 1–14, 2020, https://doi.org/10.3390/su12041646.

[5] L. Geni, E. Yulianti, and D. I. Sensuse, "Sentiment Analysis of Tweets Before the 2024 Elections in Indonesia Using IndoBERT Language Models," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 3, pp. 746–757, 2023, https://doi.org/10.26555/jiteki.v9i3.26490.

[6] M. M. Dakwah, A. A. Firdaus, F. Furizal, and R. Faresta, "Sentiment Analysis on Marketplace in Indonesia using Support Vector Machine and Naïve Bayes Method," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 10, no. 1, p. 39, 2024, https://doi.org/10.26555/jiteki.v10i1.28070.

[7] H. R. Alhakiem and E. B. Setiawan, "Aspect-Bas1ed Sentiment Analysis on Twitter Using Logistic Regression with FastText Feature Expansion," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 5, pp. 840–846, 2022, https://doi.org/10.29207/resti.v6i5.4429.

[8] R. Azizah Arilya, Y. Azhar, and D. Rizki Chandranegara, "Sentiment Analysis on Work from Home Policy Using Naïve Bayes Method and Particle Swarm Optimization," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 3, p.

433, 2021, https://doi.org/10.26555/jiteki.v7i3.22080.

[9] H. A. Santoso, E. H. Rachmawanto, A. Nugraha, A. A. Nugroho, D. R. I. M. Setiadi, and R. S. Basuki, "Hoax classification and sentiment analysis of Indonesian news using Naive Bayes optimization," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 2, pp. 799–806, 2020, https://doi.org/10.12928/TELKOMNIKA.V18I2.14744.

[10] M. T. Ahmed, M. Rahman, S. Nur, A. Z. M. T. Islam, and D. Das, "Natural language processing and machine learning based cyberbullying detection for Bangla and Romanized Bangla texts," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 20, no. 1, pp. 89–97, 2022, https://doi.org/10.12928/TELKOMNIKA.v20i1.18630.

[11] A. Ali, M. Khan, K. Khan, R. U. Khan, and A. Aloraini, "Sentiment Analysis of Low-Resource Language Literature Using Data Processing and Deep Learning," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 713–733, 2024, https://doi.org/10.32604/cmc.2024.048712.

[12] P. Sreevidya, O. V. Ramana Murthy, and S. Veni, "Sentiment analysis by deep learning approaches," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 2, pp. 752–760, 2020, https://doi.org/10.12928/TELKOMNIKA.V18I2.13912.

[13] 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, https://doi.org/10.1109/ACCESS.2019.2909919.

[14] Z. A. Khan *et al.*, "Developing Lexicons for Enhanced Sentiment Analysis in Software Engineering: An Innovative Multilingual Approach for Social Media Reviews," *Comput. Mater. Contin.*, vol. 79, no. 2, pp. 2771–2793, 2024, https://doi.org/10.32604/cmc.2024.046897.

[15] E. H. Muktafin and P. Kusrini, "Sentiments analysis of customer satisfaction in public services using K-nearest neighbors algorithm and natural language processing approach," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 1, pp. 146–154, 2021, https://doi.org/10.12928/TELKOMNIKA.V19I1.17417.

[16] M. A. Fauzi, "Random forest approach fo sentiment analysis in Indonesian language," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 1, pp. 46–50, 2018, https://doi.org/10.11591/ijeecs.v12.i1.pp46-50.

[17] K. K. Agustiningsih, E. Utami, and O. M. A. Alsyaibani, "Sentiment Analysis and Topic Modelling of The COVID-19 Vaccine in Indonesia on Twitter Social Media Using Word Embedding," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 8, no. 1, p. 64, 2022, https://doi.org/10.26555/jiteki.v8i1.23009.

[18] E. J. Caprisiano, M. H. Ramadhansyah, and A. Zahra, "Classifying possible hate speech from text with deep learning and ensemble on embedding method," *Bull. Electr. Eng. Informatics*, vol. 13, no. 3, pp. 1913–1919, 2024, https://doi.org/10.11591/eei.v13i3.6041.

[19] J. Jasmir, W. Riyadi, S. R. Agustini, Y. Arvita, D. Meisak, and L. Aryani, "Bidirectional Long Short-Term Memory and Word Embedding Feature for," *J. RESTI (Rekayasa Sist. Dan Teknol. Informasi)*, vol. 6, no. 4, pp. 505–510, 2022, https://doi.org/10.29207/resti.v6i4.4005.

[20] B. Wang, A. Wang, F. Chen, Y. Wang, and C. C. J. Kuo, "Evaluating word embedding models: Methods and experimental results," *APSIPA Trans. Signal Inf. Process.*, vol. 8, pp. 1–14, 2019, https://doi.org/10.1017/ATSIP.2019.12.

[21] P. Purwono, A. Ma'arif, I. S. Mangku Negara, W. Rahmaniar, and J. Rahmawan, "Linkage Detection of Features that Cause Stroke using Feyn Qlattice Machine Learning Model," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 3, p. 423, 2021, https://doi.org/10.26555/jiteki.v7i3.22237.

[22] S. Rapacz, P. Chołda, and M. Natkaniec, "A method for fast selection of machine-learning classifiers for spam filtering," *Electron.*, vol. 10, no. 17, 2021, https://doi.org/10.3390/electronics10172083.

[23] F. N. N. H. R. Passarella, S. Nurmaini, M. N. Rachmatullah, H. Veny, "Development of a machine learning model for predicting abnormalities of commercial airplanes," *Res. Pract. Thromb. Haemost.*, p. 100137, 2023, https://doi.org/10.1016/j.jsamd.2023.100613.

[24] Jasmir *et al.*, "Breast Cancer Classification Using Deep Learning," *Proc. Int. Conf. Electr. Eng. Comput. Sci. ICECOS,* vol. 17, pp. 237–242, 2019, https://doi.org/10.1109/ICECOS.2018.8605180.

[25] 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, https://doi.org/10.1016/j.neunet.2018.12.002.

[26] H. I. K. Fathurrahman, A. Ma'arif, and L.-Y. Chin, "The Development of Real-Time Mobile Garbage Detection Using Deep Learning," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 3, p. 472, 2022, https://doi.org/10.26555/jiteki.v7i3.22295.

[27] H. Kim, J. Kim, J. Kim, and P. Lim, "Towards perfect text classification with Wikipe dia-base d semantic Naïve Bayes learning," *Neurocomputing*, vol. 0, pp. 1–7, 2018, https://doi.org/10.1016/j.neucom.2018.07.002.

[28] J. Jasmir, S. Nurmaini, and B. Tutuko, "Fine-grained algorithm for improving knn computational performance on clinical trials text classification," *Big Data Cogn. Comput.*, vol. 5, no. 4, 2021, https://doi.org/10.3390/bdcc5040060.

[29] M. Azam, T. Ahmed, F. Sabah, and M. I. Hussain, "Feature Extraction based Text Classification using K-Nearest Neighbor Algorithm," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 12, pp. 95–101, 2018, https://doi.org/10.29207/resti.v6i4.4186.

[30] T. Salles, M. Gonçalves, V. Rodrigues, and L. Rocha, "Improving random forests by neighborhood projection for effective text classification," *Inf. Syst.*, vol. 77, pp. 1–21, 2018, https://doi.org/10.1016/j.is.2018.05.006.

[31] N. Jalal, A. Mehmood, G. S. Choi, and I. Ashraf, "A novel improved random forest for text classification using feature ranking and optimal number of trees," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2733–2742, 2022, https://doi.org/10.1016/j.jksuci.2022.03.012.

[32] Rayhan Rahmanda and Erwin Budi Setiawan, "Word2Vec on Sentiment Analysis with Synthetic Minority

Oversampling Technique and Boosting Algorithm," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 4, pp. 599–605, 2022, https://doi.org/10.29207/resti.v6i4.4186.

[33] A. George, H. B. Barathi Ganesh, M. Anand Kumar, and K. P. Soman, *Significance of global vectors representation in protein sequences analysis*, vol. 31. Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-04061-1_27.

[34] I. N. Khasanah, "Sentiment Classification Using fastText Embedding and Deep Learning Model," *Procedia CIRP*, vol. 189, pp. 343–350, 2021, https://doi.org/10.1016/j.procs.2021.05.103.

[35] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection," *Procedia Comput. Sci.*, vol. 207, no. Kes, pp. 769–778, 2022, https://doi.org/10.1016/j.procs.2022.09.132.

[36] 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, https://doi.org/10.1016/j.procs.2019.08.153.

[37] Muhammad Afif Raihan and Erwin Budi Setiawan, "Aspect Based Sentiment Analysis with FastText Feature Expansion and Support Vector Machine Method on Twitter," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 4, pp. 591–598, 2022, https://doi.org/10.29207/resti.v6i4.4187.

[38] R. Chivukula, T. Jaya Lakshmi, S. S. Uday, and S. T. Pavani, "Classifying clinically actionable genetic mutations using KNN and SVM," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 24, no. 3, pp. 1672–1679, 2021, https://doi.org/10.11591/ijeecs.v24.i3.pp1672-1679.

[39] M. S. Islam *et al.*, "Machine Learning-Based Music Genre Classification with Pre-Processed Feature Analysis," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 3, p. 491, 2022, https://doi.org/10.26555/jiteki.v7i3.22327.

[40] J. Zhang, Y. Li, F. Shen, Y. He, H. Tan, and Y. He, "Hierarchical text classification with multi-label contrastive learning and KNN," *Neurocomputing*, vol. 577, p. 127323, 2024, https://doi.org/10.1016/j.neucom.2024.127323.

[41] L. V. Nguyen, Q. T. Vo, and T. H. Nguyen, "Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services," *Big Data Cogn. Comput.*, vol. 7, no. 2, 2023, https://doi.org/10.3390/bdcc7020106.

[42] M. Nadeem *et al.*, "Preventing Cloud Network from Spamming Attacks Using Cloudflare and KNN," *Comput. Mater. Contin.*, vol. 74, no. 2, pp. 2641–2659, 2023, https://doi.org/10.32604/cmc.2023.028796.

[43] Y. Yang and X. Liu, "A re-examination of text categorization methods," *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 42-49, 1999, https://doi.org/10.1145/312624.312647.

[44] D. Berrar, "Bayes' theorem and naive bayes classifier," *Encycl. Bioinforma. Comput. Biol. ABC Bioinforma.*, vol. 1–3, no. 2018, pp. 403–412, 2018, https://doi.org/10.1016/B978-0-12-809633-8.20473-1.

[45] J. K. Alwan, D. S. Jaafar, and I. R. Ali, "Diabetes diagnosis system using modified Naive Bayes classifier," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 28, no. 3, pp. 1766–1774, 2022, https://doi.org/10.11591/ijeecs.v28.i3.pp1766-1774.

[46] S. Wang, J. Ren, and R. Bai, "A semi-supervised adaptive discriminative discretization method improving discrimination power of regularized naive Bayes," *Expert Syst. Appl.*, vol. 225, p. 120094, 2023, https://doi.org/10.1016/j.eswa.2023.120094.

[47] C. J. Anderson *et al.*, "A novel naïve Bayes approach to identifying grooming behaviors in the force-plate actometric platform," *J. Neurosci. Methods*, vol. 403, p. 110026, 2024, https://doi.org/10.1016/j.jneumeth.2023.110026.

[48] M. Badar and M. Fisichella, "Fair-CMNB: Advancing Fairness-Aware Stream Learning with Naïve Bayes and Multi-Objective Optimization," *Big Data Cogn. Comput.*, vol. 8, no. 2, 2024, https://doi.org/10.3390/bdcc8020016.

[49] A. A. Khaleel, A. A. M. Al-Azzawi, and A. M. Alkhazraji, "Random forest for lung cancer analysis using Apache Mahout and Hadoop based on software defined networking," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 32, no. 2, pp. 1086–1093, 2023, https://doi.org/10.11591/ijeecs.v32.i2.pp1086-1093.

[50] T. A. Assegie, R. Subhashni, N. K. Kumar, J. P. Manivannan, P. Duraisamy, and M. F. Engidaye, "Random forest and support vector machine-based hybrid liver disease detection," *Bull. Electr. Eng. Informatics*, vol. 11, no. 3, pp. 1650–1656, 2022, https://doi.org/10.11591/eei.v11i3.3787.

[51] A. Sekulić, M. Kilibarda, G. B. M. Heuvelink, M. Nikolić, and B. Bajat, "Random forest spatial interpolation," *Remote Sens.*, vol. 12, no. 10, pp. 1–29, 2020, https://doi.org/10.3390/rs12101687.

[52] H. Syahputra and A. Wibowo, "Comparison of Support Vector Machine (SVM) and Random Forest Algorithm for Detection of Negative Content on Websites," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 1, pp. 165–173, 2023, https://doi.org/10.26555/jiteki.v9i1.25861.

BIOGRAPHY 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.

Nurhadi is a lecture at Universitas Dinamika Bangsa Jambi, Indonesia. He received his Bachelor in Informations systems in 2004 and Master Degree in Information Technology in 2009 from Universitas Gadjah Mada Yogyakarta, Indonesia. He received a Doctor in Informatics Engineering at Universitas Sriwijaya Palembang, Indonesia in 2023. His research interest is image processing, Machine learning and multimedia. He can be contacted at email: nurhadi.rahmad06@gmail.com.

Eni Rohaini is a lecture at Universitas Dinamika Bangsa Jambi, Indonesia. She received his Bachelor in Informatics Engineering in Universitas Dinamika Bangsa Jambi in 2009 and Master Degree in Computer Science in Universitas Budi Luhur Jakarta in 2015. Her research interest is Linear Algebra, Data Mining and Information System. She can be contacted at email: enirohaini0104@gmail.com.

M. Riza Pahlevi. B is a lecture at Universitas Dinamika Bangsa Jambi, Indonesia. He received his Bachelor in Informatics Engineering in Universitas Dinamika Bangsa Jambi in 2010 and Master Degree in Computer Science in Universitas Budi Luhur Jakarta in 2015. His research interest is Linear Algebra, Data Mining and Information System. He can be contacted at email: rizapahlevikuliah@gmail.com.

Daniel Sintong Pardamean Simanjuntak is a lecture in Universitas Dinamika Bangsa**,** he received his bachelor in informatics engineering in unama jambi in 2009 and master degree in magister system information in unama, indonesia in 2013. her research interest in blockchain, data mining, Information system. he can be contacted at email: daniel.sintong.87@gmail.com.