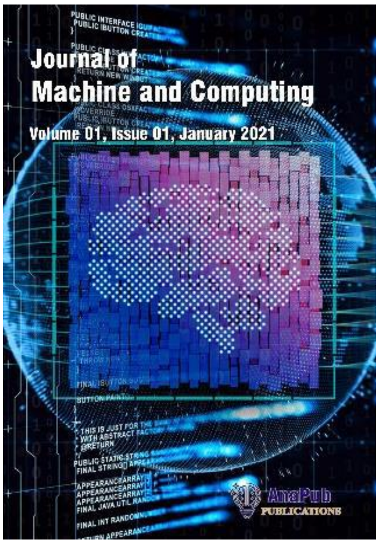
Journal Pre-proof

Influence Of Pre-Processing Strategies On Sentiment Analysis Performance: Leveraging BERT, TF-IDF and Glove Features



Kosala N and Nirmalrani V

DOI: 10.53759/7669/jmc202505036

Reference: JMC202505036

Journal: Journal of Machine and Computing.

Received 20 September 2024

Revised form 02 November 2024

Accepted 05 December 2024

Please cite this article as: Kosala N and Nirmalrani V, “Influence Of Pre-Processing Strategies On Sentiment Analysis Performance: Leveraging BERT, TF-IDF and Glove Features”, Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505036

INFLUENCE OF PRE-PROCESSING STRATEGIES ON SENTIMENT ANALYSIS PERFORMANCE: LEVERAGING BERT, TF-IDF, AND GLOVE FEATURES

N Kosala[1] and V Nirmalrani[2]

[1,2]Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.

[1]kosala.nataraj@gmail.com, [2]nirmalrani.it@sathyabama.ac.in

Correspondence should be addressed to N. Kosala: kosala.nataraj@gmail.com

ABSTRACT

The analysis of user-generated content, such as product reviews on platforms like Amazon, is critical for understanding consumer sentiment. However, the unstructured nature of these reviews poses challenges for accurate sentiment analysis(SA). This study examines the influence of different preprocessing techniques on the effectiveness of sentiment analysis utilizing three feature extraction methods: BERT, TF-IDF, and GloVe. We evaluated the effectiveness of these techniques with machine learning classifiers such as: Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), and Extreme Gradient Boosting (XGBoost). Our findings indicate that preprocessing significantly enhances classification accuracy, particularly for models using TF-IDF and GloVe features, while BERT-based models showed robust performance even with minimal preprocessing. By combining BERT with preprocessing techniques, we were able to attain an exceptional accuracy rate of 98.3% in sentiment analysis. This underscores the significance of meticulous data pretreatment in this field. These insights enhance the creation of more efficient sentiment classification algorithms, providing reliable information from Amazon product reviews.

1. INTRODUCTION

In natural language processing (NLP), SA is a fundamental method meant to extract subjective information from textual input [1]. Consumer reviews on e-commerce platforms such as Amazon offer valuable insights into product performance and customer satisfaction. This benefits consumers and encourages marketers to know consumers and their tastes, enabling them to customize their products accordingly [2]. As the number of available comments for a company increases, it becomes more difficult for potential consumers to decide whether to make a purchase [3]. In this era of artificial intelligence, it takes considerable time to categorize a sample and analyze thousands of reviews to assess a brand's appeal to customers globally [4][5].

However, these reviews are often unstructured, containing elements such as noise, emoticons, slang, and varied terminology, which complicates the analysis process. Sentiment analysis has several obstacles, one of the main challenge is informal writing style that is unstructured text.[7]The problem addressed in this study stems from the challenge of processing unstructured text to accurately classify sentiment. Unstructured Sentiment is a form of writing characterized by its casual and unrestricted nature, allowing the writer to express themselves without any imposed guidelines or constraints[8]. Pre-processing entails the removal of impurities and the conversion of the unprocessed data into a structure that is appropriate for analysis.Effective preprocessing is critical as it directly influences the performance of models. The first process in sentiment classification is to preprocess the text, transforming the unstructured data found on the web, which often contains noise, into a format suitable for classification. The next stage involves feature extraction.[6] Despite the importance of preprocessing, there is a lack of comprehensive studies comparing its impact across different feature extraction techniques, especially when using state-of-the-art models like BERT . Second step is feature extraction(FE) in SA,In our study, we utilized BERT, TF-IDF, and GloVe ,FE is an essential process in sentiment classification since it involves extracting significant information from the text input, which directly impacts the performance of the model. The approach aims to extract relevant information that encompasses the most fundamental characteristics of the text.[7].Finally machine learning algorithm is utilized to categorize sentiments.

The main contribution of this paper are:

1. The study methodically examines the influence of different preprocessing procedures on sentiment analysis performance. The research identifies the accuracy of the model before and after implementing the preprocessing techniques.

2. Next step is feature extraction process, Three prominent feature extraction method BERT, TF-IDF, and GloVe—are utilized in the study.

3. Third step is the study employs four widely-used machine learning classifiers: LR, RF, NB, and XGBoost. The performance of these classifiers is evaluated in conjunction with the different feature extraction methods and preprocessing techniques.

4. The research offers a comprehensive comparison of the classifiers performance before and after applying preprocessing techniques. This comparison highlights the Value of preprocessing in improving model dependability and accuracy.
5. This paper presents a comparative comparison of three feature extraction approaches, namely BERT, TF-IDF, and GloVe. It showcases the influence of each technique on the performance of sentiment analysis models. The study delineates the advantages and constraints of each strategy in various contexts.

The study's findings provide useful insights that can enhance the creation of sentiment analysis algorithms that are more precise and dependable. By examining the interplay between preprocessing, feature extraction, and classification, the paper offers guidance on optimizing sentiment analysis pipelines for improved performance.This study emphasizes the importance of robust pre-processing and feature extraction in SA. The results indicate that implementing suitable strategies can greatly enhance the performance of classification models, resulting in more precise and practical insights derived from user-generated material on e-commerce platforms.

2. LITERATURE REVIEW

The researchers in [9] investigated a classification algorithm for analysing the sentiment of micro-blogging posts on Twitter. By utilising several preprocessing tactics and employing numerous feature selection techniques on the Naïve Bayes classifier, the researchers achieved adequate performance on the employed training set.Ultimately, it was noted that all the trained classifiers demonstrated slightly better performance in classifying the positive class compared to the negative class. The findings indicate that by integrating the Naïve Bayes method with the utilisation of Information Gain evaluated using Chi square with a minimum threshold of 3 to choose features with high information content, an accuracy rate of 89% is achieved.
Singh and Kumari [10] assessed the impact of preprocessing strategies on Twitter data and demonstrated the enhancement of the classifiers. The URLs, hashtags, user mentions, punctuation, and stop words were eliminated, while colloquial expressions were substituted with appropriate language using n-gram techniques.
Jianqiang and Xiaolin [11] assessed these elements on five Twitter datasets by enlarging acronyms, substituting negation, eliminating URLs, numerals, and stop words.
Bao et al. [12] examined the influence of pre-processing strategies on the categorisation of sentiment in Twitter. The results suggest that incorporating the URL features reserve, negation transformation, and repeated letters normalization improves the precision of sentiment classification.
Macro et al. [13] showed that the order in which preprocessing operations are implemented significantly affects the efficacy of sentiment analysis models. The accuracy of classifiers like NB can be enhanced by employing techniques like as lemmatisation, stop word removal, and appropriate handling of negations. Effrosynidis et al [14] highlighted that applying preprocessing approaches, such as removing elongated characters, abbreviations, and misspellings, improved the accuracy by reducing unnecessary information and standardising the text.Alam et al. [15]Research has shown that preprocessing has a significant impact on machine learning algorithms. Specifically, suitable preprocessing techniques can improve the accuracy of sentiment datasets by 5%. [illegible]ika et al. [16] performed a comparative investigation of GloVe and other word embedding techniques, demonstrating a significantly high degree of precision, especially when employing the SVM method. The importance of SA, particularly on social networks, has been increasing, with BERT emerging as a crucial tool and effective technique for extracting characteristics. Kaliyar et al. [17] has demonstrated the efficacy of integrating the BERT methodology with classifiers. Their system, which combines the BERT methodology with CNN and LSTM networks, has surpassed previous algorithms in detecting false news.. The CNN model has obtained an impressive accuracy of 98.90%, while the LSTM model has reached an accuracy of 97.55%.

3. METHODOLOGY

Figure 1 depicts the sequential movement of data across multiple modules in the suggested methodology for doing SA on Amazon reviews. The design of the planned work is explained in the following sub-sections

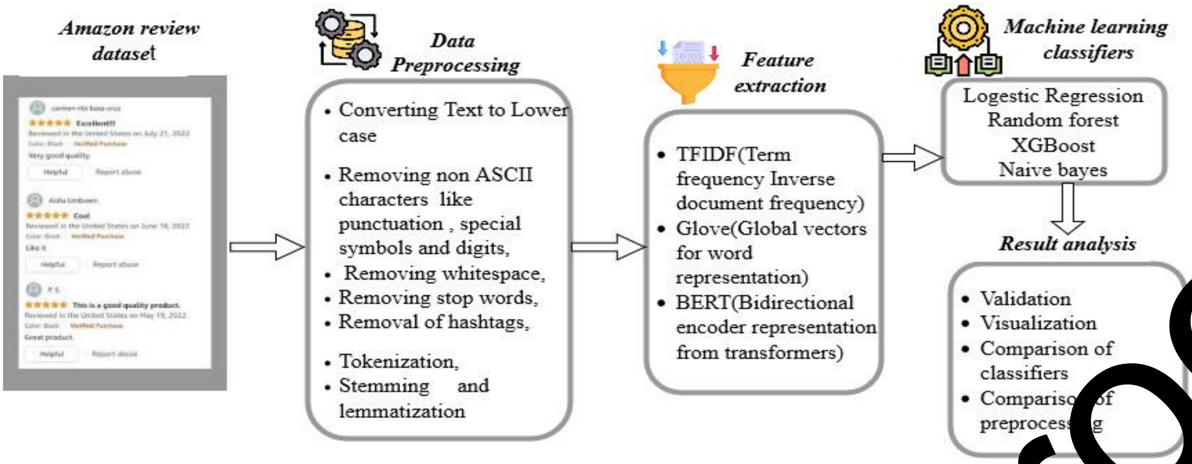


Figure 1. Architectural Data Flow Diagram of the Proposed Sentiment Analysis Framework for Amazon Product Reviews

The aim of this study was to investigate how various text preprocessing approaches affect the effectiveness of sentiment categorization models utilizing three distinct feature extraction strategies. We used the amazon product review dataset for evaluation purposes. The methodology we employ consists of the following stages: The tasks included in this project include data collection, preprocessing, feature extraction, sentiment categorisation, evaluation, and analysis.

3.1. Dataset Description

The dataset utilized is the *Consumer Reviews of Amazon Products* dataset, publicly available on Kaggle (Kaggle). It consists of 34,660 product reviews sourced from Amazon, spanning a variety of product categories, such as electronics, home goods, and apparel. The dataset was originally compiled by Datafiniti, which collected reviews from Amazon's website.

3.2. Preprocessing

Preprocessing techniques are crucial in sentiment analysis since they transform raw text input into a suitable structure for machine learning models. These tactics aim to enhance the accuracy and efficiency of sentiment categorization by refining and standardizing the data, which is especially important because of the casual style of texts from social networking platforms.

3.2.1. Stop word removal

Stop-words, such as prepositions, definite and indefinite articles, pronouns, and conjunctions, are commonly used words that provide little value in meeting an information request [18]. Eliminating these terms is a standard procedure to reduce the computing workload needed for analysis [19].

3.2.2. Lemmatization and Stemming

These are two approaches utilized in NLP to reduce words to their fundamental or core form. Nevertheless, they employ distinct methodologies and pursue marginally divergent objectives. It refers to the procedure of reducing a term to its fundamental or foundational form. The base form does not necessary need to be a linguistically valid term. Stemming algorithms, such as the Porter Stemmer, function by eliminating prevalent prefixes or suffixes from words, typically employing uncomplicated criteria.Examples: The word "running" is changed to "run",happiness"-"happi","cats"-"cat".[21] Lemmatization use vocabulary and morphological analysis to remove inflectional endings and obtain the root or canonical form of a word. The system takes into account the context and the word's part of speech in order to guarantee precision. Example: "running" → "run" ,"better" → "good" ,"geese" → "goose",It typically yields more precise and significant base forms in comparison to stemming.

3.2.3. Dealing with Abbreviations and Slang

In the field of NLP, the process of normalizing text is employed to enhance comprehension by addressing abbreviations and slang. Abbreviations are stretched to their whole forms (for example, "btw" is transformed into "by the way"), while slang phrases are substituted with their conventional counterparts (for instance, "lit" is changed to "exciting"). One way to accomplish this is by utilizing preexisting dictionaries or by employing complex NLP models that analyze the context. The objective is to enhance the coherence and facilitate the analysis of the text.

3.2.4. Eliminating Elongated Characters

Eliminating elongated characters in NLP involves reducing repeated letters in words to their standard form. For example, "soooo" is shortened to "so," and "yeeees" becomes "yes." This is important because elongated characters are often used for emphasis or expression in informal text but can cause issues in text analysis. Normalizing these words helps maintain consistency and improves the accuracy of NLP tasks.[20]

3.2.5. Punctuation Removal and Negation Handling

Although certain punctuation marks have no impact on sentiment and can be eliminated, emoticons and emojis, on the other hand, convey feeling and should be treated accordingly . It is crucial to identify and accurately handle negations since they have the ability to reverse the sentiment of a statement (for example, "not good" against "good").

3.2.6. Tokenisation

In natural language processing (NLP), tokenizing—breaking down text into smaller units—is a vital preprocessing step. Based on the particular use, these tokens might be words, subwords, characters, even sentences. Tokenizing aims mostly to simplify the language so that algorithms may examine and evaluate it more easily.Tokenization is a type of text segmentation.[22]

3.3. Feature extraction

3.3.1. TFIDF

TF-IDF is a statistical metric utilized to assess the significance of a word in a document compared to a set of documents. The product is obtained by combining two statistical measures: (TF) and (IDF). In a document, term frequency is the frequency of a term occurring.[23]

$$\mathrm{TF(t,d)} = \frac{\text{number of times t appears in d}}{\text{total number of words in d}} \qquad (1)$$

Inverse document frequency measures the word's importance inside a particular corpus. It counts the frequency of a given word among all the corpus documents.[2[illegible]]

$$\mathrm{IDF(t)} = \log \frac{\text{Total number of documents}}{\text{number of documents that contain } t} \qquad (2)$$

$$\mathrm{TF-IDF(t,d)} = \mathrm{TF(t,d)} \times \mathrm{IDF(t)} \qquad (3)$$

3.3.2. BERT

It is a deep learning architecture designed to encode textual data and pre-train text representations. It was created to overcome the difficulty of having little labelled data in natural language processing (NLP) projects [24]. Unlike Word2Vec, it operates bidirectionally, taking into account word contexts in both the forward and backward directions.As a consequence, this leads to more precise depictions of the connections between words and their contexts.BERT has gained extensive acceptance in NLP applications.The efficacy of the system is improved by training it with task-specific data and optimising its parameters. The system's great performance in numerous NLP tasks and its capacity to produce high-quality responses in natural language help to define its well-known reputation. Moreover, pre-trained BERT models' availability to the broader public makes them a preferred choice for NLP academics and practitioners [25] [26].

3.3.3. GloVe

The GloVe model is a very efficient approach that leverages global corpus statistics to optimise the learning model by considering the context window. The primary objective is to convert words into vectors and generate word vectors based on the input corpus.The implementation procedure consists of the following steps: First, a word cooccurrence matrix is constructed using the entire corpus. Next, the learning word vector is built by applying the cooccurrence matrix and the GloVe model.

The GloVe model can be represented by the subsequent equation:

$$J = \sum_{i,j}^{N} f\left(X_{ij}\right)\left(V_i^T V_j + b_i + b_j - \ln\left(X_{ij}\right)\right)^2 \qquad (4)$$

The cooccurrence matrix, denoted as X, indicates word frequency. i and j appearing together in a single window. The element Xij specifically represents the number of times this cooccurrence occurs. The window size typically ranges from 5 to 10, while Vi and Vj denote the word vectors of word i and j, respectively. bi and bj represent the deviation terms. N refers to the dimension of the cooccurrence matrix, which is N × N in size. f denotes the weight function.[1]

3.4. Classification model

This study employed four widely recognised classification models: LR, RF ,XGBoost, and NB. The selection of these models was based on their unique methods of data processing, which makes them appropriate for comparing different feature extraction strategies.

Highly efficient when there is a nearly linear connection between the characteristics and the target variable, LR is a widely used linear model estimating the likelihood of a binary outcome based on the input features. This makes it an excellent classifier for text classification tasks, serving as a solid starting point.

One kind of ensemble learning technique called RF creates several decision trees throughout the training process. It then determines the most often occurring class (for classification tasks) or the average prediction (for regression tasks) based on the outputs of these trees. RF is renowned for its resilience and capacity to accommodate a substantial amount of input features, hence mitigating the risk of overfitting.

XGBoost is a proficient and scalable implementation of gradient boosting. The algorithm produces additive trees in a sequential fashion, with each tree designed to correct the flaws committed by the previous one. It has become increasingly popular due to its high accuracy, fast processing speed, and exceptional performance, particularly in dealing with structured data and intricate relationships [28] [29].

NB A Bayesian classifier that assumes independence among predictors and uses Bayes' theorem to make probabilistic predictions.It is a highly efficient method for classifying text since it can effectively handle data with a large number of dimensions. This makes it a commonly used approach for tasks such as SA.

The classifiers were utilised on the features extracted by BERT, TF-IDF, and GloVe, with and without preprocessing, to evaluate their performance in sentiment classification. Every classifier has distinct benefits and difficulties, offering a thorough viewpoint on their appropriateness for various feature sets in SA.

4. Result and discussion

This study involved a thorough assessment of different machine learning classifiers utilising three specific methods for extracting features: BERT, TF-IDF, and GloVe. The analysis primarily aimed to evaluate the influence of preprocessing techniques on the classification performance of various models. Table 1 displays the performance comparison of classifiers utilising BERT feature with preprocessing technique. It is noteworthy that BERT with preprocessing attained the highest accuracy, reaching an impressive 98.3%

Table 1. Performance Comparison of Classifiers Using BERT Features with preprocessing

Parameter	Accuracy	Precision	Recall	F1
LR	**98**[illegible]	80	62	67
RF	97[illegible]	99	50	49
XGBoost	[illegible]7.6[illegible]	82	55	58
MNB	97.36	63	53	54

Table 2. Performance Comparison of Classifiers Using TFIDF Features with preprocessing

Parameter	Accuracy	Precision	Recall	F1
MNB	93	59	83	62
LR	93.01	60	80	64
RF	**97.5**	81	61	66
XGBoost	96.98	0.7	73	71

Table 3. Performance Comparison of Classifiers Using GLOVE Features with preprocessing

Parameter	Accuracy	Precision	Recall	F1
LR	58.19	50	49	39
RF	**97.36**	49	50	99
XGBoost	96.27	52	51	51

Table 4. Performance Comparison of Classifiers Using BERT Features without preprocessing

Parameter	Accuracy	Precision	Recall	F1
LR	**97.5**	0.83	0.56	0.59
MNB	57.2	0.51	0.61	0.4
RF	97.3	0.99	0.5	0.49
XGBoost	97.3	0.74	0.51	0.51

Table 5. Performance Comparison of Classifiers Using TFIDF Features without preprocessing

Parameter	Accuracy	Precision	Recall	F1
MNB	91.56	0.59	0.84	0.63
LR	94.19	0.63	0.84	0.68

RF	**97.12**	0.77	0.55	0.58
XGBoost	97.02	0.7	0.71	0.7

Table 6. Performance Comparison of Classifiers Using GLOVE Features without preprocessing

Parameter	Accuracy	Precision	Recall	F1
LR	59.42	0.5	0.49	0.39
RF	**97.4**	0.53	0.5	0.5
XGBoost	96.29	0.49	0.5	0.49

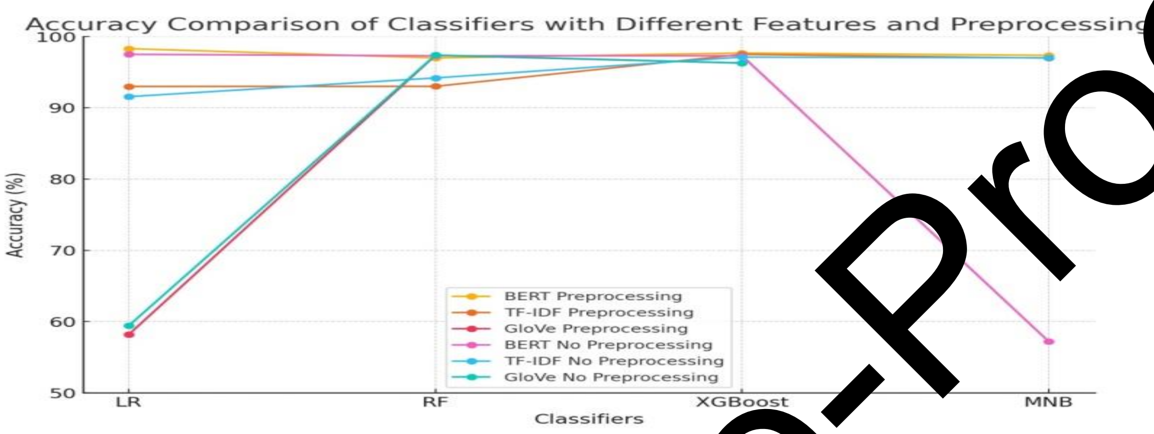


Figure 2. Shows the accuracy comparison of classifiers before and after preprocessing

The Classifiers that utilised BERT features consistently exhibited good accuracy rates, regardless of whether preprocessing was applied or not. Logistic Regression was the most successful classifier after applying preprocessing, achieving an accuracy of 98.3%. table 2 shows that TFIDF with preprocessing, the performance showed a small decrease, but it still remained strong with an accuracy of 97.5%. This demonstrates the innate robustness of BERT in capturing intricate linguistic patterns, with preprocessing providing minimal improvements.The TF-IDF feature set achieved a high level of accuracy with all classifiers, especially RF and XGBoost, both above 97% accuracy regardless of preprocessing.

In table 3,The classifiers' performance employing GloVe features exhibited the highest level of variability. After using preprocessing techniques, the RF and XGBoost models achieved impressive accuracy rates of 97.36% and 96.27% respectively. In contrast, LR had difficulties while using GloVe features, with an accuracy of only 58.19% with preprocessing and 59.42% without preprocessing. This implies that the efficacy of GloVe may rely more on the selected classifier, and some models may not fully utilise the semantic richness offered by GloVe embeddings.The effect of preprocessing is evident in the produced outcome. Preprocessing generally enhances the classification performance, particularly for models utilising TF-IDF and GloVe features. The impact of preprocessing was less noticeable in BERT-based models, which consistently achieved good performance regardless of preprocessing. This emphasises the robustness of BERT in extracting features.

Based on the classifiers analysis, XGBoost and RF were found to be the most reliable classifiers, consistently achieving high levels of accuracy across various feature extraction approaches and preprocessing circumstances. LR demonstrated robust performance, especially when used with BERT and TF-IDF characteristics. The findings indicate that whereas some classifiers, such as NB, may need preprocessing to get optimal performance, others like XGBoost and RF exhibit versatility and reliability across various feature sets.

The combination of BERT with preprocessing resulted in the best classification accuracy overall, establishing it as the most successful strategy in this investigation. Models that employed TF-IDF and GloVe approaches also achieved competitive results, especially when preprocessing procedures were applied. XGBoost and RF emerged as standout classifiers due to their constant and strong performance across different feature extraction methods, establishing them as dependable options for text classification problems.

The Comprehensive table7 comparing the net improvement in F1-Score for each classifier using different feature extraction methods (BERT, TF-IDF, and GloVe) before and after preprocessing.

Table 7. Net Improvement in F1-Score for Classifiers Before and After Preprocessing

Classifier	Feature Set	F1-Score Before Preprocessing	F1-Score After Preprocessing	Net Improvement
LR	BERT	59	67	8
RF	BERT	49	49	0
XGBoost	BERT	51	58	7
NB	BERT	40	54	14

LR	TF-IDF	68	64	-4
RF	TF-IDF	58	66	8
XGBoost	TF-IDF	70	71	1
NB	TF-IDF	63	62	-1
LR	GloVe	39	39	0
RF	GloVe	50	99	49
XGBoost	GloVe	49	51	2

Figure 3 shows the net improvement in F1-Score after preprocessing. Net Improvement refers to the change in F1-Score that occurs when preprocessing is applied, relative to the F1-Score before preprocessing.It represent the net improvement in F1-Score after preprocessing, categorized by both classifier and feature set. The chart's horizontal bars correspond to the net improvement values provided for each combination of classifier and feature set.

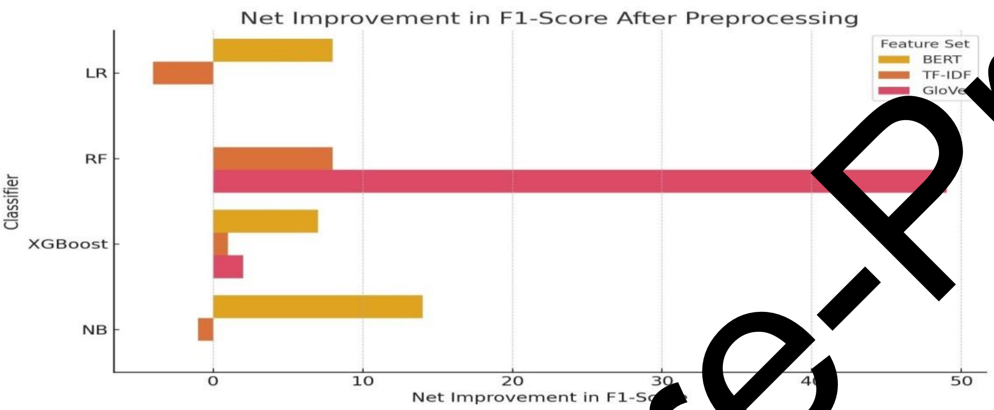


Figure 3. Shows the Net improvement in F1-Score after preprocessing.

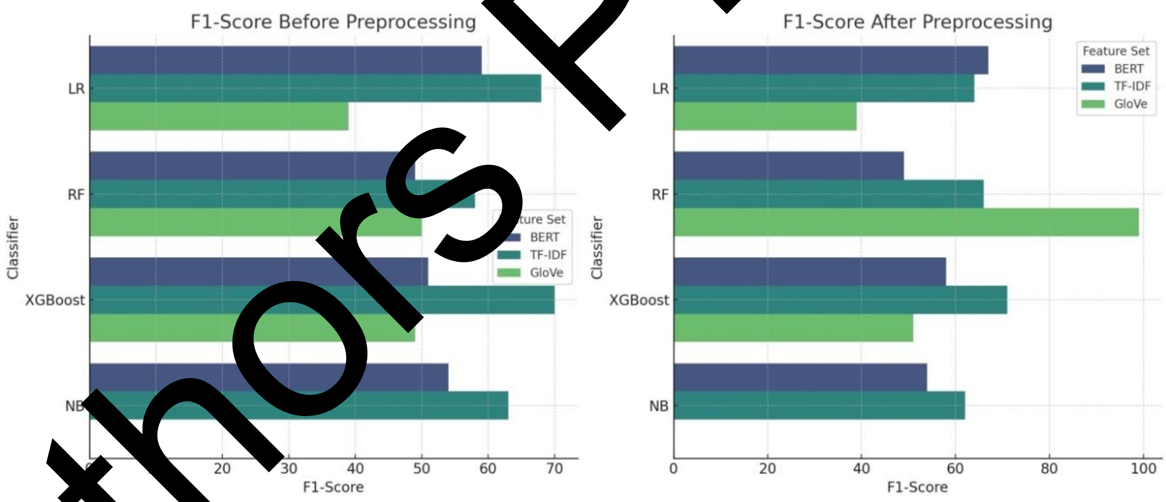


Figure 4. Comparison of F1-Scores Before and After Preprocessing Across Classifiers and Feature Sets

A positive rating signifies progress, whereas a negative value signifies a decline in performance.Feature sets in The Table 7 presents a comparison of classifiers using three distinct feature extraction methods: BERT, TF-IDF, and GloVe.The table presents a comprehensive overview of the influence of preprocessing on the effectiveness of each classifier, enabling the identification of the models that derive the most advantage from preprocessing methods. Additionally, the bar chart visually represents the net improvement in F1-Score for each classifier, categorized by the feature set used.

5. CONCLUSION

This work conducted a comprehensive comparative investigation of the efficacy of multiple machine learning classifiers in the context of SA. Three distinct feature extraction techniques, namely BERT, TF-IDF, and GloVe, were employed. The analysis specifically concentrated on the influence of preprocessing on the accuracy of the classifiers.The findings indicated that the utilisation of BERT-based features consistently resulted in the maximum accuracy for classification, while preprocessing contributed a improvement. This demonstrates the proficiency of BERT in capturing intricate linguistic patterns, rendering it a remarkably efficient technique for extracting features in SA applications. However, classifiers that used TF-IDF and GloVe also achieved good results, especially when preprocessing techniques were implemented. This highlights the significance of preprocessing in improving model performance, especially for methods that are sensitive to the distribution of features.XGBoost and RF emerged as the most dependable classifiers, continuously achieving excellent accuracy regardless of the feature extraction techniques and preprocessing settings employed. LR demonstrated robust performance, especially when utilising BERT and TF-IDF characteristics, while its efficacy varied when using GloVe.

Overall, the study found that BERT with preprocessing was the most effective approach. However, it also emphasised the need of choosing suitable preprocessing approaches and classifiers depending on the specific characteristics of the feature extraction method used. These findings offer useful insights for creating strong sentiment analysis models and highlight the need of carefully considering feature extraction and preprocessing procedures in text classification tasks.

REFERENCES

[1] L. Xiaoyan, R. C. Raga, and S. Xuemei, "GloVe-CNN-BiLSTM model for sentiment analysis on text reviews," Journal of Sensors, vol. 2022, Article ID 7212366, pp. 1-12, Oct. 2022, doi: 10.1155/2022/7212366.

[2] N. Sultan, "Sentiment analysis of Amazon product reviews using supervised machine learning techniques," Knowledge Engineering and Data Science (KEDS), vol. 5, no. 1, pp. 101-108, Dec. 2022.

[3] S. N. Ahmad and M. Laroche, "Analyzing electronic word of mouth: A social commerce construct," International Journal of Information Management, vol. 37, no. 3, pp. 202-213, Jun. 2017.

[4] Z. Xiang, Q. Du, Y. Ma, and W. Fan, "A comparative analysis of major online review platforms: Implications for social media analytics in hospitality and tourism," Tourism Management, vol. 58, pp. 51-65, Feb. 201[illegible].

[5] J. Wang, M. D. Molina, and S. S. Sundar, "When expert recommendation contradicts peer opinion: Relative social influence of valence, group identity and artificial intelligence," Computers in Human Behavior, vol. 107, p. 106278, Jun. 2020.

[6] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The impact of features extraction on sentiment analysis," in Procedia Computer Science, vol. 152, pp. 341-348, 2019, doi: 10.1016/j.procs.2019.0[illegible]72.

[7] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," Artificial Intelligence Review, vol. 55, no. 8, pp. 5731–5780, 20[illegible]2.

[8] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What Yelp fake review filter might be doing?," in Proceedings of the International AAAI Conference on Web and Social Media, vol. 7, 2013.

[9] S. F. Sayeedunnissa, A. R. Hussain, and M. A. Ham[illegible]d, "Supervised opinion mining of social network data using a bag-of-words approach on the cloud," in 7th International Conference on Bio-Inspired Computing: Theories and Application, 2013.

[10] T. Singh and M. Kumari, "Role of text pre-processing in Twitter sentiment analysis," Procedia Computer Science, vol. 89, pp. 549–554, 2016.

[11] J. Zhang and X. Geng, "Comparison research on text pre-processing methods on Twitter sentiment analysis," IEEE Access, vol. 5, pp. 2870–2879, 2017.

[12] Y. Bao, C. Quan, L. Wang, and F. Ren, "The role of pre-processing in Twitter sentiment analysis," in Proc. 10th Int. Conf. (ICIC), Taiyuan, China, Apr. 2014, pp. 615–[illegible].

[13] M. A. Palomino and F. Aider, "Evaluating the effectiveness of text pre-processing in sentiment analysis," Applied Sciences, 2022.

[14] D. Effrosynidis, S. Symeonidis, and A. Arampatzis, "A comparison of pre-processing techniques for Twitter sentiment analysis," in TPDL, 2017.

[15] S. Alam and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis," Computational and Mathematical Organization Theory, 2019.

[16] S. Sagnika, B. S. P. Mishra, and [illegible]. K. Meher, "Improved method of word embedding for efficient analysis of human sentiments," Multimedia Tools and Applications, vol. 79, no. 43–44, pp. 32389–32413, 2020, doi: 10.1007/s11042-020-09632-9.

[17] R. [illegible] Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," Multimedia Tools and Applications, vol. 80, pp. 11765–11788, doi: 10.1007/s11042-020-10183-2.

[18] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge University Press, 2008.

[19] [illegible] P. Sinka and D. Corne, "Evolving better stoplists for document clustering and web intelligence," in Design and Application of Hybrid Intelligent Systems, IOS Press, 2003, pp. 1015–1023.

[20] R. Lourdusamy and S. Abraham, "A Survey on Text Pre-processing Techniques and Tools," International Journal of Computer Sciences and Engineering, vol. 6, Special Issue-3, pp. 148-157, Apr. 2018.

[21] A. [illegible] Kadhim, "An Evaluation of Preprocessing Techniques for Text Classification," International Journal of Computer Science and Information Security (IJCSIS), vol. 16, no. 6, pp. 22-32, June 2018.

[22] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," Information Processing & Management, vol. 50, no. 1, pp. 104-112, Jan. 2014, doi: 10.1016/j.ipm.2013.08.006.

[23] M. Avinash and E. Sivasankar, "A Study of Feature Extraction Techniques for Sentiment Analysis," in Advances in Intelligent Systems and Computing, vol. 814, pp. 475-486, Sept. 2018. doi: 10.1007/978-981-13-1501-5_41.

[24] K. Purwandari, T. W. Cenggoro, J. W. C. Sigalingging, and B. Pardamean, "Twitter-based classification for integrated source data of weather observations," IAES International Journal of Artificial Intelligence (IJ-AI), vol. 12, no. 1, pp. 271–283, Mar. 2023, doi: 10.11591/ijai.v12.i1.pp271-283.

[25] A. Maiti, A. Abarda, and M. Hanini, "The impact of feature extraction techniques on the performance of text data classification models," Indonesian Journal of Electrical Engineering and Computer Science, vol. 35, no. 2, pp. 1041–1052, Aug. 2024, doi: 10.11591/ijeecs.v35.i2.pp1041-1052.

[26] R. Nareshkumar and K. Nimala, "Interactive Deep Neural Network for Aspect-Level Sentiment Analysis," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), pp. 1–8, Jan. 2023, doi: 10.1109/iceconf57129.2023.10083812.

[27] H. A and B. S. P, “Deep Learning with Crested Porcupine Optimizer for Detection and Classification of Paddy Leaf Diseases for Sustainable Agriculture,” Journal of Machine and Computing, pp. 1018–1031, Oct. 2024, doi: 10.53759/7669/jmc202404095.
[28] S. Nithya and S. U. -, “MOOC Dropout Prediction using FIAR-ANN Model based on Learner Behavioral Features,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 9, 2022, doi: 10.14569/ijacsa.2022.0130972.
[29] P. V, U. S, S. B, R. V, M. Thangaraju, and U. M. P, “Advanced Explainable AI: Self Attention Deep Neural Network of Text Classification,” Journal of Machine and Computing, pp. 586–593, Jul. 2024, doi: 10.53759/7669/jmc202404056.