Influence of Pre-Processing Strategies on Sentiment Analysis Performance: Leveraging Bert, TF-IDF and Glove Features

[1]Kosala N and [2]Nirmalrani V
[1,2]Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India.
[1]kosala.nataraj@gmail.com, [2]nirmalrani.it@sathyabama.ac.in

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

Article Info
Journal of Machine and Computing (https://anapub.co.ke/journals/jmc/jmc.html)
Doi : https://doi.org/10.53759/7669/jmc202505036
Received 18 September 2024; Revised from 30 October 2024; Accepted 18 November 2024.
Available online 05 January 2025.

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 attained 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.

Keywords – Preprocessing Techniques, Text Embedding Techniques, BERT, TFIDF, GLOVE, Sentiment Analysis, Machine Learning Classifiers, Amazon Reviews.

I. 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, varying review lengths, and inconsistent grammatical structures**,** which complicate the analysis process. For example, colloquial language, abbreviations (e.g., "btw" for "by the way"), and emojis can carry nuanced meanings that are difficult for models to interpret without proper preprocessing. Additionally, reviews range from a few words ("Loved it!") to lengthy paragraphs detailing multiple aspects of a product, further increasing variability in the dataset. Sentiment analysis (SA) faces several obstacles due to the informal writing style**s** prevalent in user-generated content. Unstructured sentiment is a form of writing characterized by its casual and unrestricted nature, allowing users to express themselves without any imposed guidelines or constraints [7]. This lack of standardization introduces challenges in extracting meaningful features for classification.

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]. To address these issues, preprocessing entails the removal of impurities such as stopwords, punctuation, and irrelevant characters while handling elements like emoticons and slang to ensure accurate representation. For example, the phrase "This product is lit 🔥!" requires text normalization to standardize "lit" as "exciting" and interpret the emoji's sentiment. By converting raw, unstructured text into a cleaner, analyzable format, preprocessing plays a critical role in directly influencing the performance of sentiment analysis models.

The first step in sentiment classification is to preprocess the text, transforming the unstructured data often found on platforms like Amazon into a structured format suitable for classification. This preprocessing bridges the gap between noisy user-generated content and the structured data required by machine learning algorithms, thereby enhancing the effectiveness of sentiment analysis pipelines.

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 contributions of this paper are:

- Investigates the impact of various preprocessing techniques on the performance of sentiment analysis models. And provides a comparative analysis of model accuracy before and after the application of preprocessing methods.
- The next phase involves the feature extraction process, where three prominent methods **BERT**, **TF-IDF**, and **Glove**—are employed to extract meaningful features for sentiment analysis.
- Employed and evaluated the performance of four widely used machine learning classifiers L**R**, **RF**, **NB**, and **XGBoost** in conjunction with feature extraction methods (**BERT**, **TF-IDF**, and **GloVe**) and various preprocessing techniques.
- Comparison of classifier performance is conducted to evaluate their results before and after applying preprocessing techniques. This analysis highlights the critical role of preprocessing in improving the dependability and accuracy of sentiment analysis models.
- Finally, this paper presents a comparative analysis of three feature extraction approaches—**BERT**, **TF-IDF**, and **GloVe**. It examines the impact of each technique on the performance of sentiment analysis models and outlines the advantages and limitations of each method in different 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.

II. 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.

In [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.

The [11] assessed these elements on five Twitter datasets by enlarging acronyms, substituting negation, eliminating URLs, numerals, and stop words. The [12] examined the influence of pre-processing strategies on the categorisation of sentiment in Twitter. The results suggest that incorporating the URL feature reserve, negation transformation, and repeated letters normalization improves the precision of sentiment classification.

In [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. [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. [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%.

In [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. The research [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%, while the LSTM model has reached an accuracy of 97.55%.

III. METHODOLOGY

Fig 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.

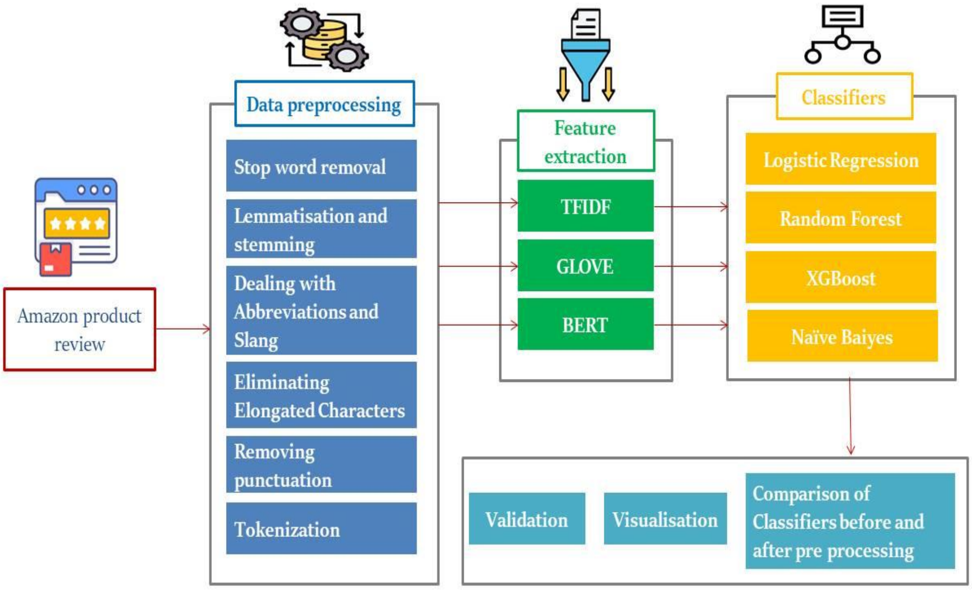

Fig 1. Architectural Diagram of the Proposed Sentiment Analysis Framework for Amazon Product Reviews.

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 employed consists of the following stages: data collection, preprocessing, Feature extraction, sentiment Categorization, evaluation and analysis.

Dataset Description

This project leverages a dataset obtained from Kaggle.com, focusing on Amazon product reviews. The dataset encompasses a substantial collection of over 34,600 reviews contributed by customers across diverse product categories, including electronics, home furniture, and various other commodities. Beyond customer reviews, the dataset incorporates crucial elements such as product ratings and a diverse set of additional information. The dataset includes comprehensive details ranging from product specifications to star ratings provided by customers, encompassing a holistic perspective of customer feedback and product attributes.

Preprocessing

Preprocessing techniques are crucial in sentiment analysis since they transform raw text input into a suitable structure for machine learning models. These steps aim to enhance the accuracy and efficiency of sentiment classification by refining and standardizing the data, which is especially important given the casual style of texts from social networking platforms. The preprocessing pipeline used in this study includes the following steps:

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 reduces noise and decreases the computational workload required for analysis [19]. Some examples of English stop words include "the," "she," "us," "we," "her," and "himself." For instance, "This is a great product" becomes "great product" after stop-word removal.

Lemmatisation 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. lt considers the context and part of speech, yielding more accurate root forms (e.g., "better" → "good", "geese" → "goose").", This step enhances text representation for sentiment analysis

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, ensuring that informal expressions are transformed into their formal equivalents. For instance, "btw this product is lit" is normalized to "by the way this product is exciting." that is 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.

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]

Punctuation Removal and Negation Handling

Although certain punctuation marks have no impact on sentiment and can be eliminated, Punctuation marks, such as commas and periods, are removed to reduce noise. However, special characters like hashtags, @mentions, and emojis are treated appropriately to preserve sentiment. For example:

- "#amazing" becomes "amazing."
- Emojis such as 😃 are mapped to their corresponding sentiment (e.g., "happy").

Additionally, negations are handled to ensure correct sentiment interpretation (e.g., "not good" vs. "good").

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]. For example, the sentence "This product is amazing!" is tokenized into ["This", "product", "is", "amazing"]. This step ensures that the input is prepared for feature extraction techniques like TF-IDF, GloVe, and BERT.

Feature Extraction

Feature extraction is a pivotal step in natural language processing (NLP), transforming textual data into numerical representations that machine learning algorithms can process. In this study, we independently evaluated three distinct feature extraction methods: TF-IDF, GloVe, and BERT. Each method offers unique advantages and caters to different aspects of text representation.

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]

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

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.[23]

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

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

BERT (Bidirectional Encoder Representation from Transformers)

BERT is a pre-trained language model created by Devlin et al. to enhance the quality and efficiency of NLP solutions. Additionally, it has been effectively employed in several NLP tasks, including question answering and text classification.

It is engineered to comprehend the contextual meaning of words within a phrase by analysing them bidirectionally—from left to right and right to left. This feature renders BERT exceptionally proficient for text representation jobs, such as sentiment analysis, where comprehending subtle links between words is essential.
BERT employs WordPiece Tokenisation, which deconstructs text into smaller subword units to manage infrequent or out-of-vocabulary terms. For instance, the term "unhappiness" may be divided into "un," "##happy," and "##ness." This subword tokenisation allows BERT to accurately represent unusual words while maintaining contextual significance. special tokens in BERT [CLS] Represents the complete sequence and is utilised for sentence-level tasks (e.g., sentiment categorisation).[SEP] Distinguishes sentences or phrases in tasks such as question responding or subsequent sentence prediction.For instance, the phrase "The product is amazing!" is tokenised as: [CLS]The product is amazing! [SEP]

Every token is represented as a composition of the subsequent embeddings:

- Token embedding: a word embedding represented as a numerical vector.
- Segment embedding: the static representation that determines whether a token is part of the first or second sentence when the latter is provided as input.
- Position embedding: the positional representation for each character within a sentence. In the case of two sentences, the position in the second sentence follows sequentially from the final position of the first sentence, incremented by one.

BERT is built on the **Transformer architecture**, consisting of the following key components[11],First **Encoder Layers** , BERT uses multiple transformer encoders (e.g., 12 in BERT_BASE), Each layer processes the input sequence to capture word relationships.Then **Multi-Head Attention Mechanism**,Captures the importance of each word in the sequence relative to other words, allowing the model to focus on relevant parts of the input.Next **Feed-Forward Neural Networks**,refines the contextual representation of words.Finally **Hidden Layers**, Each encoder layer outputs a 768-dimensional vector for each token (in BERT_BASE).

For sentiment analysis, the [CLS] token's final hidden state is often used as a feature representation for the entire input sequence. This vector captures the context and meaning of the input text. First process is Tokenize the text into WordPiece tokens then Pass the tokenized input through BERT and Extract the [CLS] token's output as the feature vector for the input text. finally Use this vector as input to a downstream classifier (e.g., Logistic Regression, Random Forest) to predict sentiment.

Global Vectors for Word Representation

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:

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

- X_{ij}*: Frequency of co-occurrence between words i and j*

The co-occurrence matrix, represented as X, captures the frequency with which words appear together within a given context. The context window size generally ranges between 5 and 10 words. In this matrix, Vi and Vj denote the vector representations of words i and j, respectively, while bi and bj are the associated bias terms. The matrix has dimensions N×N where N corresponds to the size of the vocabulary. The weight function f, is used to assign importance to the co-occurrence values [1].

Rationale for Method Selection

The selection of TF-IDF, GloVe, and BERT for independent evaluation is motivated by their distinct approaches to text representation,**TF-IDF** Captures the importance of words based on their frequency, effectively identifying keywords and sentiment-laden terms,**GloVe** Encodes semantic relationships by analyzing global co-occurrence statistics, facilitating the understanding of word similarities and differences,**BERT** Provides deep contextual embeddings, comprehending word meanings in relation to their surrounding context, which is crucial for nuanced sentiment analysis.Evaluating these methods independently allows for a comprehensive comparison of statistical, semantic, and contextual feature extraction techniques, offering valuable insights into their individual strengths and limitations in sentiment analysis tasks.In **Suitability for the Dataset** The Amazon reviews dataset presents challenges such as informal language, slang, and varying review lengths. Each method addresses these challenges differently, **TF-IDF** Effectively highlights frequently occurring sentiment-heavy terms. **GloVe** Captures semantic similarities, aiding in recognizing synonyms and antonyms. **BERT** Handles informal writing styles, abbreviations, and emojis through its contextual embeddings. The independent evaluation of these methods revealed that **TF-IDF**: Performs well with explicit sentiment terms but may struggle with context-dependent phrases.

GloVe: Effectively captures semantic relationships but faces challenges with out-of-vocabulary words and slang. **BERT**: Excels in handling informal and complex language but requires higher computational resources.

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.

LR: 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.

RF: 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.

NB: Naïve Bayes (NB) classifies an instance according to the Bayesian theorem of conditional probability. The probabilities of an instance belonging to each of the c_k classes given the instance x is denoted as $P(c_k|x)$. Naïve Bayes classifiers argue that, conditional on the class variable, the value of a specific feature is independent of the values of all other features.

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.

IV. 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	**98.3**	80	62	67
RF	97.0	99	50	49
XGBoost	97.68	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**	83	56	59
MNB	57.2	51	61	40
RF	97.3	99	50	49
XGBoost	97.3	74	51	51

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

Parameter	Accuracy	Precision	Recall	F1
MNB	91.56	59	84	63
LR	94.19	63	84	68
RF	**97.12**	77	55	58
XGBoost	97.02	70	71	70

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

Parameter	Accuracy	Precision	Recall	F1
LR	59.42	50	49	39
RF	**97.4**	53	50	50
XGBoost	96.29	49	50	49

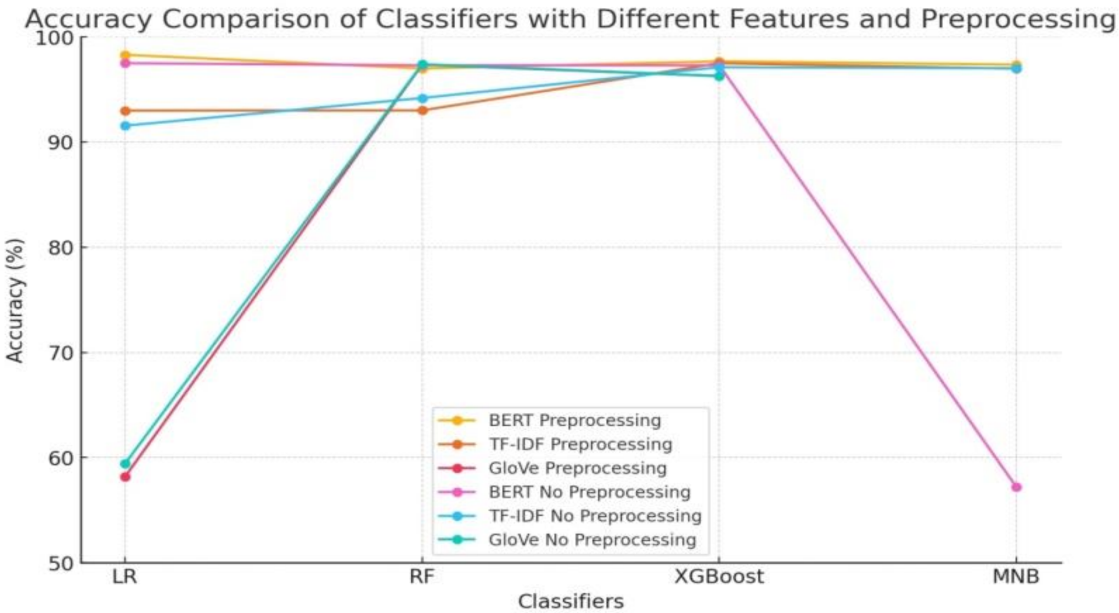

Fig 2. Shows the Accuracy Comparison Of Classifiers Before And After Preprocessing.

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. **Fig 2** Shows the Accuracy Comparison Of Classifiers Before And After 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. **Table 4** shows Performance Comparison of Classifiers Using BERT Features without Preprocessing.

Based on the classifier's 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. **Table 5** shows Performance Comparison of Classifiers Using TFIDF Features without Preprocessing.

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. **Table 6** shows Performance Comparison of Classifiers Using GLOVE Features without Preprocessing.

The Comprehensive **Table 7** 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

Fig 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.

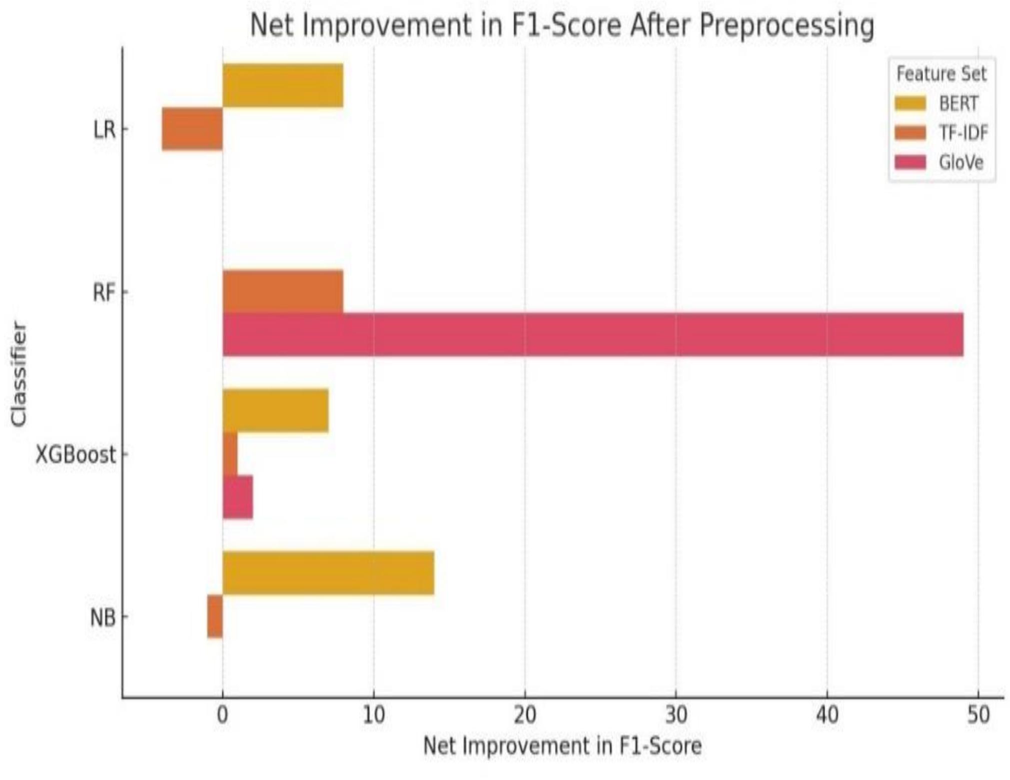

Fig 3. Shows the Net Improvement In F1 Score After Preprocessing.

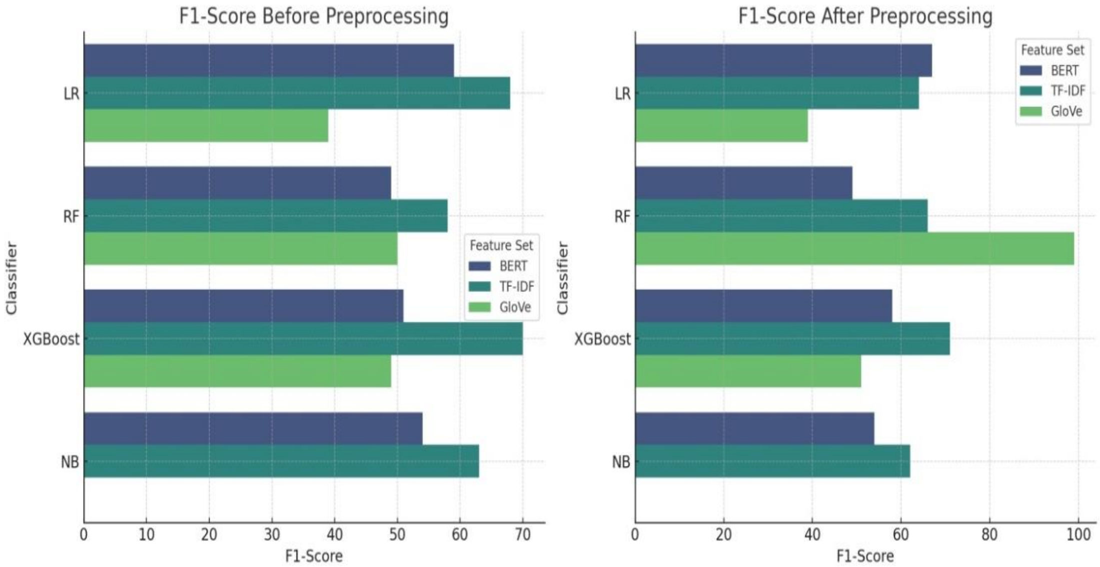

Fig 4. Comparison of F1-Scores Before and After Preprocessing Across Classifiers and Feature Sets.

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. **Fig 4** shows Comparison of F1-Scores Before and After Preprocessing Across Classifiers and Feature Sets.

V. 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 with preprocessing consistently resulted in the maximum accuracy with 98.3% for classification. 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. In future work, combining multiple preprocessing strategies could be explored to further enhance the quality of data preparation. Additionally, experimenting with diverse feature extraction techniques, such as integrating traditional and deep learning-based methods, could provide new insights and potentially boost overall performance in sentiment analysis tasks

CRediT Author Statement

The authors confirm contribution to the paper as follows:
Conceptualization: Kosala N and Nirmalrani V; **Methodology:** Kosala N and Nirmalrani V; **Writing- Original Draft Preparation:** Kosala N; **Investigation:** Nirmalrani V; **Supervision:** Nirmalrani V; All authors reviewed the results and approved the final version of the manuscript.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests

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, 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, vol. 5, no. 1, p. 101, Jun. 2022, doi: 10.17977/um018v5i12022p101-108.
[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, doi: 10.1016/j.ijinfomgt.2016.08.004.
[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. 2017, doi: 10.1016/j.tourman.2016.10.001.
[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, doi: 10.1016/j.chb.2020.106278.
[6]. R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The Impact of Features Extraction on the Sentiment Analysis," Procedia Computer Science, vol. 152, pp. 341–348, 2019, doi: 10.1016/j.procs.2019.05.008.
[7]. M M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," Artificial Intelligence Review, vol. 55, no. 7, pp. 5731–5780, Feb. 2022, doi: 10.1007/s10462-022-10144-1.
[8]. A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What Yelp Fake Review Filter Might Be Doing?," Proceedings of the International AAAI Conference on Web and Social Media, vol. 7, no. 1, pp. 409–418, Aug. 2021, doi: 10.1609/icwsm.v7i1.14389.
[9]. S. Fouzia Sayeedunnissa, A. R. Hussain, and M. A. Hameed, "Supervised Opinion Mining of Social Network Data Using a Bag-of-Words Approach on the Cloud," Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), pp. 299–309, Dec. 2012, doi: 10.1007/978-81-322-1041-2_26.
[10]. T. Singh and M. Kumari, "Role of Text Pre-processing in Twitter Sentiment Analysis," Procedia Computer Science, vol. 89, pp. 549–554, 2016, doi: 10.1016/j.procs.2016.06.095.
[11]. Z. Jianqiang and G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," IEEE Access, vol. 5, pp. 2870–2879, 2017, doi: 10.1109/access.2017.2672677.
[12]. Y. Bao, C. Quan, L. Wang, and F. Ren, "The Role of Pre-processing in Twitter Sentiment Analysis," Intelligent Computing Methodologies, pp. 615–624, 2014, doi: 10.1007/978-3-319-09339-0_62.
[13]. M. A. Palomino and F. Aider, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis," Applied Sciences, vol. 12, no. 17, p. 8765, Aug. 2022, doi: 10.3390/app12178765.
[14]. D. Effrosynidis, S. Symeonidis, and A. Arampatzis, "A Comparison of Pre-processing Techniques for Twitter Sentiment Analysis," Research and Advanced Technology for Digital Libraries, pp. 394–406, 2017, doi: 10.1007/978-3-319-67008-9_31.
[15]. R. Krishnan and S. Durairaj, "Reliability and performance of resource efficiency in dynamic optimization scheduling using multi-agent microservice cloud-fog on IoT applications," Computing, vol. 106, no. 12, pp. 3837–3878, Jun. 2024, doi: 10.1007/s00607-024-01301-1.
[16]. S. Sagnika, B. S. P. Mishra, and S. 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, Aug. 2020, doi: 10.1007/s11042-020-09632-9.
[17]. M. P. Sinka and D. Corne, "Evolving better stoplists for document clustering and web intelligence," in Design and Application of Hybrid Intelligent Systems, IOS Press, pp. 1015–1023, 2003.
[18]. R. Lourdusamy and S. Abraham, "A Survey on Text Pre-processing Techniques and Tools," International Journal of Computer Sciences and Engineering, vol. 06, no. 03, pp. 148–157, Apr. 2018, doi: 10.26438/ijcse/v6si3.148157.
[19]. I. 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.
[20]. 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.
[21]. M. Avinash and E. Sivasankar, "A Study of Feature Extraction Techniques for Sentiment Analysis," Emerging Technologies in Data Mining and Information Security, pp. 475–486, Sep. 2018, doi: 10.1007/978-981-13-1501-5_41.
[22]. Devlin J, Chang M-W, Lee K, Toutanova K, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, arXiv preprint arXiv:1810.04805.
[23]. C. Sun, L. Huang, & X. Qiu, "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence, (2019), arXiv preprint arXiv:1903.09588.