Enhancing Sentiment Analysis with Feature Extraction and Dimensionality Reduction in Traditional Machine Learning Models

Emanuele Bossi[1] and Faisal Ahmed[1]

Embry-Riddle Aeronautical University, Prescott AZ 86301, USA,
bossie@my.erau.edu, ahmedf09@erau.edu

Abstract. This study investigates the role of feature extraction and dimensionality reduction in enhancing sentiment analysis performance. We explore the effectiveness of different feature extraction techniques, such as Bag of Words and Term Frequency-Inverse Document Frequency, and apply a simple statistical feature selection method, the Chi-Square test, to improve model efficiency. The models evaluated in this study include Naive Bayes and Logistic Regression, both of which are white-box models that offer interpretability and computational efficiency. Using the IMDB movie reviews dataset, we found that by carefully selecting features and utilizing higher-order n-grams, our models achieved accuracy levels comparable to those obtained by more complex models, such as BERT. Additionally, the simplicity of the models enabled faster training and inference times, processing the data in just a few seconds compared to the several hours required by deep learning models. The results demonstrate that traditional machine learning approaches, when combined with effective feature extraction and dimensionality reduction techniques, can yield high-performance results while maintaining transparency and efficiency.

Keywords: Sentiment Analysis, Dimensionality Reduction, Chi-Square Test, Feature Selection, Interpretability

1 Introduction

Sentiment analysis, also known as opinion mining, is the computational task of determining the sentiment expressed in a piece of text, typically classifying it as positive, negative, or neutral [18]. With the rapid growth of online platforms such as social media, product reviews, and forums, sentiment analysis has become a critical tool in various domains, including business, politics, and healthcare [13]. It enables organizations to gain valuable insights into public opinion, customer satisfaction, and emerging trends. The increasing volume and variety of text data make sentiment analysis a challenging task, as it requires not only accurate text processing techniques but also an understanding of the context in which sentiment is expressed.

As the volume and diversity of text data grow, one of the key challenges in sentiment analysis is managing the high dimensionality of feature spaces. Textual data, when transformed into numerical representations, often leads to large and sparse feature sets, which can overwhelm models and degrade their performance. Effective feature selection and dimensionality reduction techniques are critical for addressing these challenges, ensuring that sentiment analysis models remain both computationally efficient and accurate.

While recent advancements in natural language processing (NLP) and machine learning have significantly improved the accuracy and efficiency of sentiment analysis models, with techniques such as deep learning, transformer-based architectures like BERT (Bidirectional Encoder Representations from Transformers) [16], and recurrent neural networks (RNNs) [22] that have shown substantial promise in handling the complexities of language and sentiment expression, traditional machine learning models such as Naïve Bayes and Logistic Regression remain highly relevant due to their simplicity, interpretability, and scalability. These models, however, rely heavily on the quality and relevance of the features extracted from the text.

This paper focuses on the role of feature selection in reducing dimensionality and improving model performance in sentiment analysis. Specifically, we investigate the impact of univariate statistical techniques (e.g., Chi-Square) as feature selection methods on two machine learning algorithms: Multinomial Naïve Bayes (MNB) and Logistic Regression (LR). Through systematic experiments, we aim to identify the optimal balance between feature set size and classification accuracy, highlighting the importance of dimensionality reduction in enhancing the efficiency and generalizability of sentiment analysis models.

By emphasizing the critical role of feature selection and dimensionality reduction, this study contributes to the development of efficient and robust sentiment analysis pipelines. The findings provide practical insights for researchers and practitioners seeking to optimize traditional machine learning models for text classification tasks in high-dimensional spaces.

2 Related Work

Sentiment Analysis (SA) has been an active area of research for over two decades, and various techniques have been proposed to address its challenges. These techniques can be broadly categorized into three groups: lexicon-based approaches, machine learning-based methods, and deep learning-based techniques. This section provides a review of the most widely used techniques in sentiment analysis and their effectiveness across various tasks.

2.1 Lexicon-Based Approaches

Lexicon-based methods rely on predefined sentiment lexicons that associate words or phrases with their sentiment scores (positive, negative, or neutral). One of the earliest and most commonly used lexicons is the AFINN lexicon, which assigns sentiment scores to words based on their polarity [14]. Similarly, SentiWordNet is another popular lexicon that enriches WordNet with sentiment-related information, providing a useful resource for analyzing textual data in terms of sentiment orientation [4].

Although lexicon-based methods are simple and interpretable, they often suffer from limitations such as the inability to handle context and the challenge of dealing with domain-specific language. To overcome some of these issues, researchers have proposed hybrid models that combine lexicons with other techniques, such as rule-based systems or machine learning classifiers [8].

2.2 Machine Learning-Based Approaches

Machine learning approaches have been widely used for sentiment classification, particularly after the widespread availability of large annotated corpora for training. Commonly used algorithms include Support Vector Machines (SVMs), Naïve Bayes (NB), and Decision Trees (DT), all of which rely on handcrafted features derived from text, such as n-grams, part-of-speech tags, and syntactic structures.

Among these, Support Vector Machines (SVMs) have been shown to perform well for sentiment classification tasks due to their ability to handle high-dimensional data and effectively separate sentiment classes in a feature space [27]. In comparison, Naïve Bayes classifiers, though simpler, are often used due to their efficiency and ease of implementation. While these methods have yielded solid results in many sentiment analysis tasks, they still face challenges related to feature extraction and the sparsity of high-quality labeled data [3].

In recent years, ensemble techniques, such as Random Forests and boosting algorithms like AdaBoost and Gradient Boosting Machines (GBM), have been proposed to further improve performance by combining multiple base models [2]. These ensemble methods typically outperform single classifiers, especially when dealing with noisy or imbalanced data, a common issue in sentiment analysis.

2.3 Deep Learning-Based Approaches

The advent of deep learning has significantly advanced the field of sentiment analysis, particularly with the introduction of neural networks that can automatically learn features from raw text. Early deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), demonstrated promising results in sentiment classification by capturing complex patterns in textual data [24], [1].

Long Short-Term Memory (LSTM) networks, a type of RNN, have gained popularity for sentiment analysis due to their ability to capture long-range dependencies and contextual information in sentences. LSTM-based models have been shown to outperform traditional machine learning classifiers in various sentiment analysis benchmarks, particularly for tasks involving longer text sequences, such as product reviews or social media posts [5]. Similarly, Gated Recurrent Units (GRUs), another variation of RNNs, have also been successfully applied to sentiment analysis tasks [17].

The introduction of Transformer models, such as BERT (Bidirectional Encoder Representations from Transformers), has marked a significant breakthrough in the field. BERT-based models, which utilize attention mechanisms to capture contextual relationships between words, have consistently outperformed previous methods on sentiment analysis benchmarks, particularly in tasks requiring fine-grained understanding of sentence-level sentiment [12]. These models, often fine-tuned for specific sentiment analysis tasks, have achieved state-of-the-art results in multiple languages and domains, setting new standards for sentiment classification accuracy.

3 Methodology

This study investigates the role of feature selection in enhancing dimensionality reduction for sentiment analysis. Specifically, we examine how dimensionality reduction techniques impact the accuracy of different models applied to the IMDB movie reviews dataset. The methodologies explored in this study aim to assess how different feature extraction and reduction approaches influence model accuracy and efficiency in sentiment analysis. The following sections describe the specific techniques employed, as shown in the flowchart displayed in Figure 1.

3.1 Dataset

The IMDB dataset contains 50,000 labeled movie reviews, evenly divided into 25,000 positive and 25,000 negative reviews, making it an ideal resource for binary sentiment classification tasks. This balanced composition ensures that both sentiment categories are equally represented, reducing bias in training and evaluation. Additionally, the reviews cover a diverse range of movie genres and user opinions, providing a comprehensive benchmark for testing sentiment analysis models.

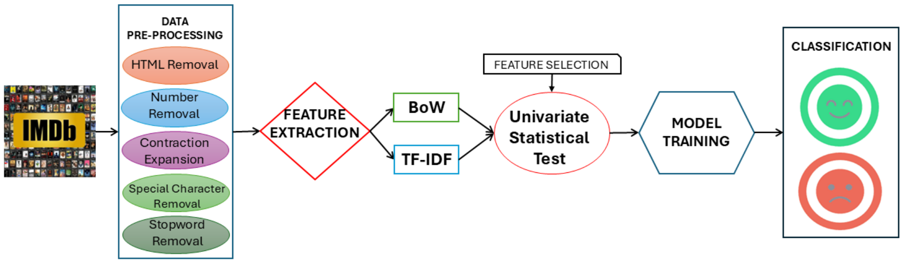

Fig. 1. Flowchart displaying the various steps performed in the execution of the experiment.

3.2 Data Preprocessing

Data preprocessing plays an essential role in improving the accuracy and efficiency of sentiment analysis models, as it directly affects the quality and consistency of the input data [23]. Raw text data is often noisy and inconsistent, and effective preprocessing helps remove irrelevant information while retaining meaningful features.

We applied several preprocessing techniques and evaluated their impact on model performance using Naive Bayes and Logistic Regression models trained with unigrams Bag of Words (BoW). The preprocessing steps we tested include:

1. **HTML Removal:** HTML tags are often included in web-scraped text but do not contribute to sentiment analysis. Removing them helps focus the model on the actual content of the review [19].
2. **Number Removal:** Numbers can be extraneous, especially in a movie review context, where they may not provide meaningful sentiment signals. Removing them can reduce the feature space and improve the model's focus on relevant textual data [26].
3. **Extra Space Removal:** Extra spaces or whitespace can introduce noise, leading to incorrect tokenization. Removing them ensures that tokenization works effectively and consistently across the dataset.
4. **Expanding Contractions:** Expanding contractions (e.g. ”don't” ”do not”) helps standardize text, which can improve model performance by reducing variability in word forms.
5. **Special Character Removal:** Special characters such as punctuation and non-alphanumeric symbols do not typically carry sentiment meaning and are removed to prevent them from being misinterpreted by the model.
6. **Stopword Removal:** Stopwords like ”the,” ”is,” and ”and” are often discarded since they are common and contribute little to sentiment classification. However, we decided to modify the stopword list from NLTK's library by retaining certain words that we believe carry sentiment-relevant information (e.g., ”not,” ”no,” ”never,” ”but,” ”more,” ”very,” and ”just”) [9].
7. **Lemmatization:** Lemmatization reduces words to their base or root form, improving consistency and ensuring that variations of a word (e.g., ”running,” ”ran”) are treated as the same word [21].

We evaluated the impact of various pre-processing modifications on the IMDB dataset by applying them and assessing the performance of Naive Bayes and Logistic Regression models. The results, summarized in Table 1, illustrate how each pre-processing technique influenced model accuracy compared to the baseline performance achieved using raw, unprocessed data. The baseline accuracy scores for the models on raw data were 84.42 for Naïve Bayes and 88.53 for Logistic Regression.

By comparing the accuracy scores obtained with and without pre-processing, we identified techniques that consistently improved model performance. Based on this analysis, we selected several pre-processing steps for our study, including HTML removal, number removal, contraction expansion, special character

Table 1. Data Preprocessing Evaluation.

Technique	**Accuracy**	
	Naïve Bayes	***Logistic Regression***
HTML Removal	84.47	88.83
Number Removal	84.50	88.50
Extra Space Removal	84.42	88.53
Expanding Contractions	84.45	88.80
Special Char Removal	84.85	88.74
Stopword Removal	85.79	88.22
Lemmatization	84.36	88.37

removal, and stopword removal using a modified NLTK list. These steps were chosen to standardize the data and enhance the models' ability to extract meaningful patterns from the reviews.

3.3 Feature Extraction

Feature extraction is a crucial step in sentiment analysis, as it transforms raw text data into numerical representations that machine learning models can process. In this study, we apply several widely-used methods to convert the text into a form that captures both the frequency and context of words, ensuring that the features are informative and discriminative for sentiment classification tasks. The following methods were applied:

- **Bag of Words (BoW)**: Bag of Words is one of the most straightforward feature extraction methods. In BoW, each document is represented as a vector, with each dimension corresponding to a unique word in the entire corpus [6]. The value of each dimension represents the frequency (or presence) of the word in the document. This representation is sparse and ignores the order of words, focusing solely on the occurrence of words as features. For a given document d, the BoW vector $\mathbf{v_d}$ can be defined as:

$$\mathbf{v_d} = [f(w_1, d), f(w_2, d), \ldots, f(w_n, d)] \tag{1}$$

  Where:
  - $w_1, w_2, \ldots, w_n$ are the words (or terms) in the vocabulary (corpus).
  - $f(w_i, d)$ is the frequency (or presence) of the word w_i in the document d.

  For this study, we applied BoW using unigrams, bigrams, and trigrams. Unigrams capture individual words, bigrams capture consecutive word pairs, and trigrams capture word triplets. Higher-order n-grams (bigrams and trigrams) help capture more context and phrases that may be important for sentiment analysis, such as "not good" or "very bad," which may convey important sentiment information not captured by unigrams alone. However, while BoW is simple and effective, it can lead to high-dimensional sparse vectors and fails to capture semantic relationships between words.

– **Term Frequency-Inverse Document Frequency**: TF-IDF improves upon the BoW model by adjusting the raw frequency of words with their Inverse Document Frequency (IDF). This approach emphasizes words that are frequent in a document but rare across the entire corpus, helping to capture more discriminative features [11]. TF-IDF provides a weighted representation of words, allowing words that are important for a specific document but not common across all documents to carry more weight in the model. For a term t in a document d, the term frequency is:

$$TF(t,d) = \frac{Number\ of\ times\ t\ appears\ in\ d}{Total\ number\ of\ terms\ in\ d}$$

The IDF for a term t is given by:

$$IDF(t) = \log(\frac{N}{df(t)})$$

where
- N is the total number of documents in the corpus.
- $df(t)$ is the number of documents that contain the term t.

The TF-IDF score for a given term is given by:

$$TF - IDF = TF\,(t,d) \times IDF\,(t).$$

As with BoW, we applied unigrams, bigrams, and trigrams to the TF-IDF model. By considering both the term frequency (TF) and document frequency (DF) of words, TF-IDF reduces the influence of common words (e.g., “the,” “and,” “is”) that do not contribute to sentiment classification, and highlights more informative words. This helps improve the model’s ability to distinguish between positive and negative sentiments.

As illustrated above, we applied Bag of Words (BoW) and TF-IDF with unigrams, bigrams, and trigrams to the preprocessed IMDB dataset. The resulting sparse matrix for each n-gram of words has the dimensionality proprieties showed in Table 2.

Table 2. Dimension Analysis.

Feature Extraction Method	**n-Gram Configuration**	**Number of Unique Features**	**Sparsity**
BoW	Unigrams	213,101	0.9995
	Bigrams	3,292,753	0.9999
	Trigrams	8,961,544	0.9999
TF-IDF	Unigrams	213,101	0.9995
	Bigrams	3,292,753	0.9999
	Trigrams	8,961,544	0.9999

For each model and feature extraction combination, we observed the results displayed in Table 3 and 4.

Table 3. BoW Evaluation Results

Feature Extraction	**Accuracy**	
	Naïve Bayes	***Logistic Regression***
Unigrams	85.81	88.19
Bigrams	88.52	89.79
Trigrams	88.82	89.70

Table 4. TF-IDF Evaluation Results

Feature Extraction	**Accuracy**	
	Naïve Bayes	***Logistic Regression***
Unigrams	86.41	88.82
Bigrams	88.90	88.18
Trigrams	89.05	87.32

The results from the Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction methods reveal important insights into the impact of n-gram selection on sentiment analysis performance using Naive Bayes and Logistic Regression models. As illustrated in Table 3 and Table 4, both models demonstrated a clear preference for specific n-gram configurations, with performance variations between unigrams, bigrams, and trigrams.

In the BoW evaluation, we observe that Naive Bayes performance improved with increasing n-gram size. The accuracy for unigrams was 85.81, which increased to 88.52 with bigrams, and further to 88.82 with trigrams. This trend indicates that Naive Bayes benefits from capturing higher-order word dependencies, with trigrams providing the highest accuracy. However, for Logistic Regression, bigrams yielded the best performance, achieving an accuracy of 89.79, slightly outperforming trigrams (89.70). This suggests that Logistic Regression performs optimally with a balanced context from consecutive word pairs, while the addition of trigrams offers minimal improvement.

The TF-IDF evaluation shows a different pattern. For Naive Bayes, accuracy was highest with trigrams (89.05), followed by bigrams (88.90), and unigrams (86.41). This indicates that Naive Bayes benefits more from the weighted term frequencies in TF-IDF, where higher-order n-grams (such as trigrams) provide more discriminative power. Conversely, Logistic Regression saw a decline in performance with trigrams (87.32), after reaching its peak with unigrams (88.82). The drop in performance with trigrams suggests that the added complexity of trigrams, when weighted by TF-IDF, may introduce noise and diminish the model's ability to generalize effectively.

When comparing the two feature extraction methods, TF-IDF generally yielded better results than BoW for Naive Bayes, particularly with trigrams, where it achieved the highest accuracy (89.05 compared to 88.82). In contrast, Logistic Regression performed slightly better with BoW (89.79 for bigrams) than with TF-IDF (88.82 for unigrams), indicating that BoW might be more suited to capture the necessary features for this model when working with simple word frequencies, without the added complexity of term weighting.

3.4 Feature Selection

To analyze the impact of feature selection on model performance, we applied Univariate Statistical Tests using the Chi-Square (X^2) test. The goal was to evaluate how the number of features influences classification accuracy by selecting the top k features based on their relevance to the target variable [28]. The Chi-Square test measures the dependence between each feature and the class label. For a feature x_i and class c, the test statistic is calculated as:

$$X^2 = \sum_j \frac{(O_{ij}-E_{ij})^2}{E_{ij}}$$

where O_{ij} is the observed frequency and E_{ij} is the expected frequency under the assumption of independence. Features with higher X^2 scores are considered more relevant.

The Chi-Square test was applied to both BoW and TF-IDF feature sets to select the most relevant features and evaluate the resulting performance. For each model and feature extraction combination, we determined the optimal number of features to select by analyzing the accuracy towards different number of features selected.

Table 5 and Table 6 summarize the results after selecting the optimal number of features ("k") for each feature extraction method, demonstrating how dimensionality reduction can effectively balance computational efficiency and classification accuracy.

Table 5. BoW Evaluation Results Using Feature Selection

Feature Extraction	**Naïve Bayes**		**Logistic Regression**	
	Optimal k	***Accuracy***	***Optimal k***	***Accuracy***
Unigrams	40,000	87.48	10,000	88.57
Bigrams	300,000	93.43	300,000	90.74
Trigrams	550,000	95.01	1,000,000	91.15

As shown in the Table 5 and Table 6, the impact of feature selection on model performance was substantial. For Naive Bayes with BoW, the trigram configuration achieved the highest accuracy of 95.01, improving significantly from 88.82 without feature selection. Logistic Regression also benefited, with the optimal trigram configuration yielding an accuracy of 91.15 compared to 89.70 without feature selection. A similar trend was observed for TF-IDF, where feature selection improved accuracy for Naive Bayes across all n-gram configurations, with the highest accuracy of 91.80 achieved using trigrams. Logistic Regression,

Table 6. TF-IDF Evaluation Results Using Feature Selection.

Feature Extraction	**Naive Bayes**		**Logistic Regression**	
	Optimal k	***Accuracy***	***Optimal k***	***Accuracy***
Unigrams	50000	88.29	50000	89.18
Bigrams	300000	91.65	1500000	88.74
Trigrams	1500000	91.80	700000	87.96

however, exhibited a marginal decline in performance with feature selection for trigrams (87.96) compared to the unoptimized result (87.32).

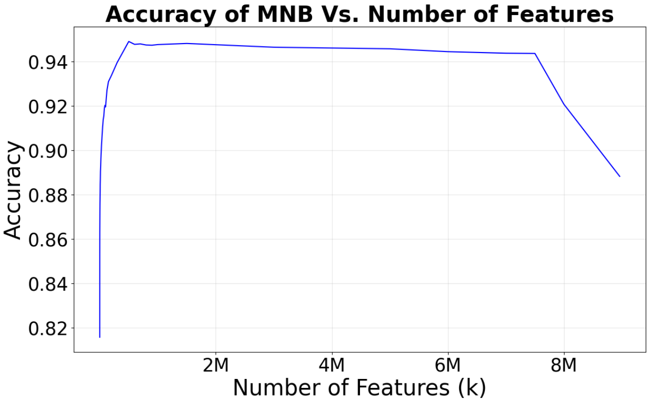

Fig. 2. Accuracy of the MNB model as a function of the number of features using Bag of Words with trigrams. The plot demonstrates that optimal accuracy (95.01) is achieved with 550,000 features. Beyond this point, the accuracy remains stationary initially, followed by a noticeable decline as the number of features increases further.

The analysis highlights the dual benefits of feature selection: reducing computational complexity by eliminating redundant or irrelevant features and enhancing model accuracy by retaining the most informative ones. Importantly, these reductions transformed the feature spaces from prohibitively high dimensions into manageable subsets without compromising model performance. For Naive Bayes, which is particularly sensitive to feature sparsity, feature selection yielded the most pronounced improvements, especially for configurations involving higher-order n-grams, as shown in 7.

The findings, moreover, further indicate that while both models benefit from dimensionality reduction, the degree of improvement varies. Logistic Regression, being less sensitive to sparsity, demonstrated less dramatic gains compared to Naive Bayes. Additionally, the optimal feature subset sizes differed significantly

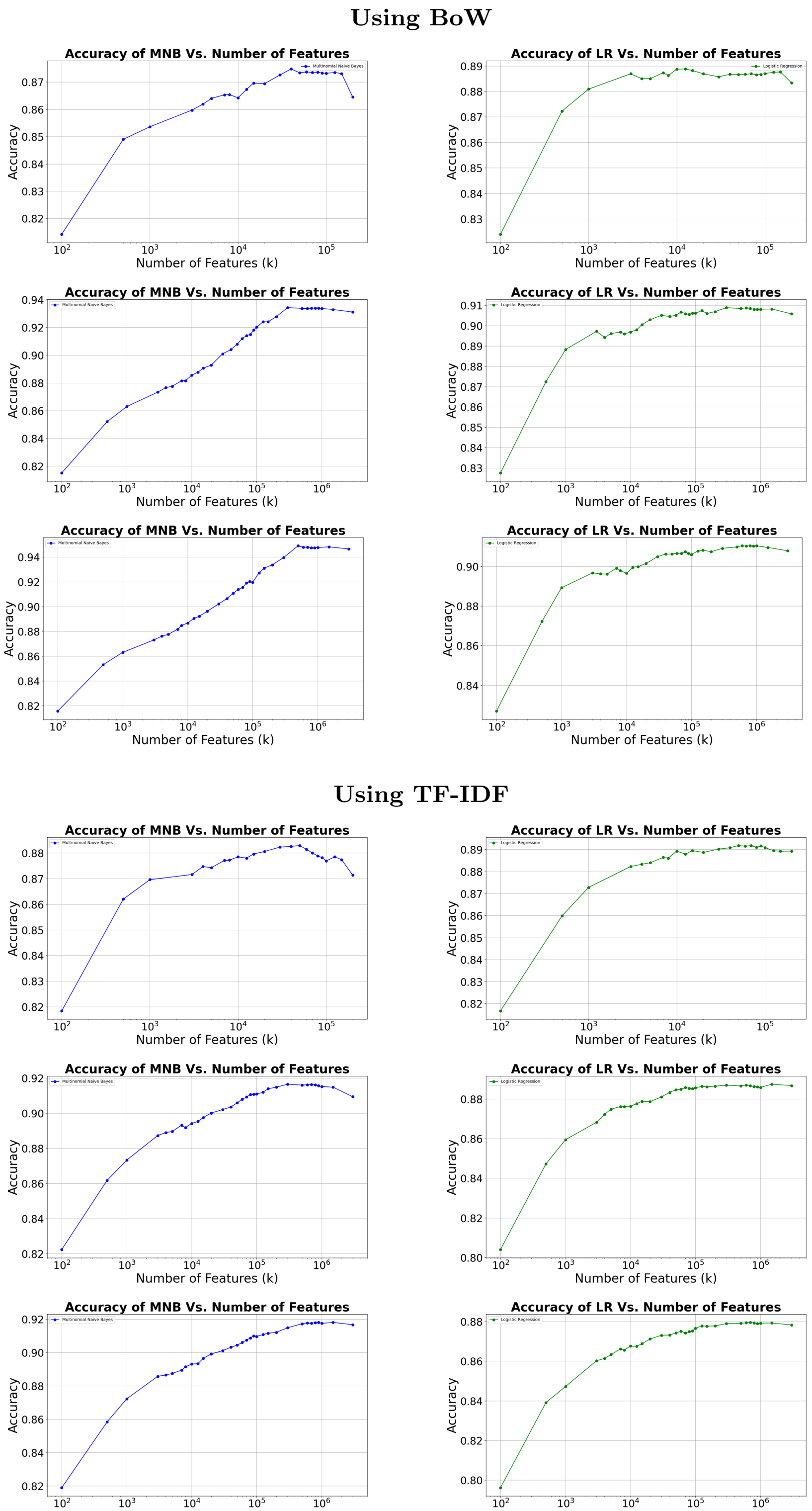

Fig. 3. Accuracy of each model-feature extraction combination as a function of the number of features (x-axis displayed on a logarithmic scale). The plots illustrate that for each case, an optimal number of features is reached, beyond which no further improvements in accuracy are observed, underscoring the importance of feature selection for effective dimensionality reduction.

between the models, reflecting their differing requirements for effective generalization.

4 Experiments

4.1 Machine Learning Models

In this study, we employed two machine learning algorithms for sentiment analysis: Multinomial Naïve Bayes (MNB) and Logistic Regression (LR). Both models were evaluated on the IMDB dataset using the Bag of Words (BoW) feature extraction method with unigrams, bigrams, and trigrams. Additionally, we explored how the selection of features impacted model accuracy by applying Univariate Statistical Tests to reduce the feature space and analyzing the results.

Multinomial Naive Bayes (MNB)
The MNB classifier is well-suited for text classification tasks, particularly when working with discrete features such as word counts or term frequencies. It assumes that features are conditionally independent given the class label and follows a multinomial distribution. The conditional probability for classifying a document d with n features $x = (x_1, x_2, \ldots,\ x_n)$ is given by:

$$P(c|x) \propto P(c) \prod_{i=1}^{n} P(x_i|c)$$

where $P(c)$ is the prior probability of class c, and $P(x_i|c)$ is the likelihood of feature x_i given the class c. The model predicts the class $\hat{c}$ that maximizes $P(c|x)$ [25].

Logistic Regression
Logistic Regression is a discriminative model that estimates the probability of a class label y given the input features x using the sigmoid function [10]. The probability $P(y = 1|x)$ is expressed as:

$$P\left(y = 1|x\right) = \frac{1}{1+e^{-(w^T x + b)}}$$

where w is the weight vector, b is the bias term, and $w^T x$ is the dot product of the feature vector and the weights. The model optimizes the weights by minimizing the binary cross-entropy loss function:

$$Loss = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(P(y_i)) + (1 - y_i) \log(1 - P(y_i))]$$

where N is the number of training samples, y_i is the true label, and $P\left(y_i\right)$ is the predicted probability.

4.2 Experimental setup

The dataset was split into training and testing subsets with an 80:20 ratio, where 80% of the data was allocated for training the machine learning models, and the remaining 20% was reserved for testing to assess the models' ability to generalize. This division ensures that the models are trained on a sufficiently large portion of the data while maintaining an independent test set for unbiased performance evaluation. To further investigate the effect of different train-test splits on model performance, Figure 4 illustrates how accuracy varies with different ratios. As seen in the plot, the accuracy remains consistently stable across all test sizes, indicating that the proposed model can effectively learn from the data regardless of the training set size.

We utilized the Chi-Square test as a feature selection method to identify the most relevant features for the classification task. The Chi-Square test measures the dependence between each feature and the target variable, evaluating whether the occurrence of a feature is independent of the class label. By selecting features that exhibit a strong dependence on the target variable, we can reduce the feature space, minimizing noise and enhancing the model's ability to generalize.

For dimensionality reduction, we applied the SelectKBest method, which is a wrapper around the Chi-Square test for feature selection. This method selects the top k features that have the highest Chi-Square statistics, where k is the number of features to retain. The optimal value of k was determined empirically by testing different values and analyzing their impact on the model's performance.

To determine the optimal number of features to retain, we systematically evaluated the performance of both models (Naive Bayes and Logistic Regression) across different values of k. The performance was assessed using cross-validation on the training set, with accuracy being the primary metric. We performed experiments for several values of k and recorded the results in Figure 3.

In Figure 3, the performance curves (accuracy vs. number of selected features) demonstrate how the accuracy of the models varied as k changed. The optimal value of k was selected based on the point at which accuracy was maximized without overfitting the model. This value represented the best trade-off between reducing the feature space and maintaining predictive power.

No data augmentation techniques were employed in this experiment. The models were trained solely on the original dataset, as the feature selection process was sufficient to improve performance by reducing dimensionality. Augmenting the data was not considered necessary, as Chi-Square feature selection already helped mitigate issues related to noise and irrelevant features.

4.3 Computational Complexity & Runtime

The experiments were conducted on a MacBook Air 2020 laptop with 8GB RAM and Intel i5 processor. The code was implemented using Python with key libraries such as scikit-learn for machine learning, NumPy for numerical operations, and pandas for data manipulation.

The reviews contained in the IMDb dataset were preprocessed and vectorized into a feature matrix; depending on the preprocessing method used (such as Bag-of-Words or TF-IDF) and the configuration employed (such as unigram, bigrams, and trigrams), the number of features typically ranges from 213,101 to 8,961,544 tokens. This dataset is sufficiently large to capture meaningful patterns in sentiment, yet manageable enough to not require specialized hardware like GPUs or distributed computing resources for model training and evaluation, especially using the machine learning methodologies employed by this study.

The training time of the models primarily depends on the number of features and the size of the dataset. For both Naive Bayes and Logistic Regression, the training process scales with the number of reviews and the size of the feature matrix. The Chi-Square feature selection technique was applied to reduce the dimensionality of the feature space, which helped speed up the training process by focusing only on the most relevant features for sentiment classification. In particular, the total runtime for the model with the best accuracy score was 42.5097 seconds, including training, testing, and feature selection.

5 Results and Analysis

This section presents the experimental results evaluating the impact of feature extraction and dimensionality reduction on sentiment analysis performance, with comparisons to findings from previously published studies. The analysis focuses on demonstrating how the method developed in this study affects model performance and classification accuracy.

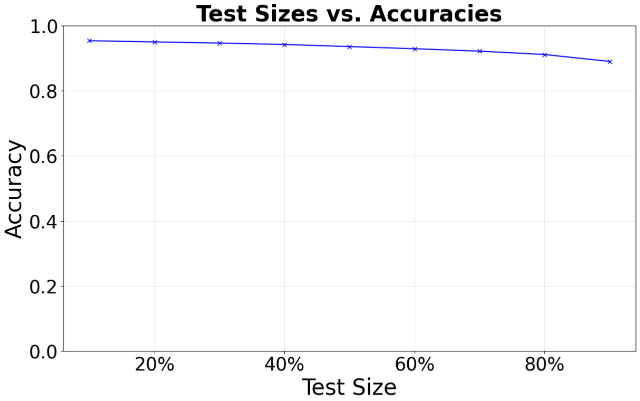

Fig. 4. Accuracy of the best model as a function of the training-testing ratio.

The Table 7 highlights the accuracy improvement after applying the Chi-Square (χ^2) test for feature selection in Naïve Bayes and Logistic Regression models using Bag of Words (BoW) and TF-IDF. Naïve Bayes showed substantial gains, with trigrams using BoW achieving the highest improvement (6.97%). Logistic Regression exhibited smaller increases, with maximum improvement (1.62%) for trigrams with BoW. Overall, higher-order n-grams and the BoW approach benefited most from the Chi-Square test, particularly in Naïve Bayes, emphasizing its effectiveness in enhancing model accuracy.

Table 7. Improvement in Accuracy Score After Applying Chi-Square Test

Model	**Feature Extraction**		**Baseline Accuracy**	**Accuracy After X^2**	**Improvement**
Naïve Bayes	BoW	Unigrams	85.81	87.48	1.95%
		Bigrams	88.52	93.43	5.55%
		Trigrams	88.82	95.01	6.97%
	TF-IDF	Unigrams	86.41	88.29	2.18%
		Bigrams	88.90	91.65	3.09%
		Trigrams	89.05	91.80	3.08%
Logistic Regression	BoW	Unigrams	88.19	88.57	0.43%
		Bigrams	89.79	90.74	1.06%
		Trigrams	89.70	91.15	1.62%
	TF-IDF	Unigrams	88.82	89.18	0.41%
		Bigrams	88.18	88.74	0.64%
		Trigrams	87.32	87.96	0.73%

Table 8. Performance Comparison of Our Model and State-of-the-Art Deep Learning Models on the IMDb Dataset

Paper	**Model**	**Accuracy**
LlamBERT: Large-scale low-cost data annotation in NLP [7]	RoBERTa-large	96.68
Supervised and Semi-Supervised Text Categorization using LSTM for Region Embeddings [15]	oh-2LSTMp	94.06
Performance Analysis of Different Neural Networks for Sentiment Analysis on IMDb Movie Reviews [11]	CNN	90.00
Sentiment analysis deep learning model based on a novel hybrid embedding method [20]	GRU+WordFast	89.54
Our Paper	**Chi-Square+Naïve Bayes**	**95.01**

Recent studies on sentiment analysis using the IMDB dataset have shown notable advancements in model performance. Table 8 provides a summary of the accuracies achieved by some of the top-performing models reported in the literature. From Table 8, it is evident that the best-performing model in this study, which employs a Naïve Bayes classifier with Bag of Words-trigrams for feature extraction (selecting 550,000 features using the Univariate Statistical Test, i.e., Chi-Square), rivals the performance of more complex models such as those based on large language models (LLMs), like BERT. In fact, this model outperforms all other "simple" machine learning models reviewed in the literature. Notably, while models like BERT can take several hours to train, the proposed Naïve Bayes model in this study is trained in just a few seconds, demonstrating its efficiency without sacrificing accuracy.

6 Conclusions

This study demonstrates that simple machine learning models like Naïve Bayes and Logistic Regression, combined with effective feature extraction (BoW and TF-IDF) and dimensionality reduction, can achieve accuracy comparable to complex models like BERT. Our approach offers significant advantages in computational efficiency, completing tasks in seconds compared to hours for BERT, while maintaining interpretability. The Chi-Square Test for feature selection significantly enhanced model performance, with a 6.97% accuracy improvement for the best model. These results highlight the critical role of dimensionality reduction in high-sparsity datasets and reaffirm the value of traditional models for applications balancing accuracy, efficiency, and explainability.

7 Future Works

While this study provides promising results, there are several directions for future research. One potential avenue is to explore more advanced feature selection techniques, such as mutual information or L1 regularization, to further improve performance. Another interesting direction would be to extend the current work to more complex datasets and multilingual sentiment analysis tasks to evaluate the generalizability of the findings.

8 Acknowledgement

This work was supported and funded by the College of Arts & Sciences at Embry-Riddle Aeronautical University, Prescott (AZ). We would like to express our sincere gratitude to Dr. Hatahet, Dean of the College, and Dr. Fisher, Chair of the Department of Mathematics.

Bibliography

[1] Qurat Tul Ain, Mubashir Ali, Amna Riaz, Amna Noureen, Muhammad Kamran, Babar Hayat, and A Rehman. Sentiment analysis using deep learning techniques: a review. *International Journal of Advanced Computer Science and Applications*, 8(6), 2017.

[2] Oscar Araque, Ignacio Corcuera-Platas, J Fernando Sánchez-Rada, and Carlos A Iglesias. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236–246, 2017.

[3] Muhammad Zubair Asghar, Aurangzeb Khan, Shakeel Ahmad, and Fazal Masud Kundi. A review of feature extraction in sentiment analysis. *Journal of Basic and Applied Scientific Research*, 4(3):181–186, 2014.

[4] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani, et al. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204. Valletta, 2010.

[5] Burhan Bilen and Fahrettin Horasan. Lstm network based sentiment analysis for customer reviews. *Politeknik Dergisi*, 25(3):959–966, 2021.

[6] Shrnivas Biradar, GT Raju, and KM Divakar. Negation handling and domain generalization in sentiment analysis using machine learning models. In *2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, volume 1, pages 1–4. IEEE, 2024.

[7] Bálint Csanády, Lajos Muzsai, Péter Vedres, Zoltán Nádasdy, and András Lukács. Llambert: Large-scale low-cost data annotation in nlp. *arXiv preprint arXiv:2403.15938*, 2024.

[8] Cach N Dang, María N Moreno-García, and Fernando De la Prieta. Hybrid deep learning models for sentiment analysis. *Complexity*, 2021(1):9986920, 2021.

[9] Kranti Vithal Ghag and Ketan Shah. Comparative analysis of effect of stopwords removal on sentiment classification. In *2015 international conference on computer, communication and control (IC4)*, pages 1–6. IEEE, 2015.

[10] Pramod Gupta and Naresh K Sehgal. *Introduction to machine learning in the cloud with python: Concepts and practices*. Springer Nature, 2021.

[11] Md Rakibul Haque, Salma Akter Lima, and Sadia Zaman Mishu. Performance analysis of different neural networks for sentiment analysis on imdb movie reviews. In *2019 3rd International conference on electrical, computer & telecommunication engineering (ICECTE)*, pages 161–164. IEEE, 2019.

[12] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. Aspect-based sentiment analysis using bert. In *Proceedings of the 22nd nordic conference on computational linguistics*, pages 187–196, 2019.

[13] Doaa Mohey El-Din Mohamed Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*, 30(4):330–338, 2018.

[14] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225, 2014.

[15] Rie Johnson and Tong Zhang. Supervised and semi-supervised text categorization using lstm for region embeddings. In *International Conference on Machine Learning*, pages 526–534. PMLR, 2016.

[16] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota, 2019.

[17] Nicole Kai Ning Loh, Chin Poo Lee, Thian Song Ong, and Kian Ming Lim. Mpnet-grus: Sentiment analysis with masked and permuted pre-training for language understanding and gated recurrent units. *IEEE Access*, 2024.

[18] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.

[19] Mayuri Mhatre, Dakshata Phondekar, Pranali Kadam, Anushka Chawathe, and Kranti Ghag. Dimensionality reduction for sentiment analysis using pre-processing techniques. In *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, pages 16–21. IEEE, 2017.

[20] Chafika Ouni, Emna Benmohamed, and Hela Ltifi. Sentiment analysis deep learning model based on a novel hybrid embedding method. *Social Network Analysis and Mining*, 14(1):210, 2024.

[21] Vinay Kumar Pant, Rupak Sharma, and Shakti Kundu. An overview of stemming and lemmatization techniques. *Advances in Networks, Intelligence and Computing*, pages 308–321.

[22] Alpna Patel and Arvind Kumar Tiwari. Sentiment analysis by using recurrent neural network. In *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*, 2019.

[23] Saurav Pradha, Malka N Halgamuge, and Nguyen Tran Quoc Vinh. Effective text data preprocessing technique for sentiment analysis in social media data. In *2019 11th international conference on knowledge and systems engineering (KSE)*, pages 1–8. IEEE, 2019.

[24] Maryem Rhanoui, Mounia Mikram, Siham Yousfi, and Soukaina Barzali. A cnn-bilstm model for document-level sentiment analysis. *Machine Learning and Knowledge Extraction*, 1(3):832–847, 2019.

[25] UK Balaji Saravanan, M Vijay, T Shreedhar, G Rajasekar, R Yashwanth, and P Shakthipriya. Multinomial naive bayes based machine learning analysis of twitter sentiment. In *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, pages 429–434. IEEE, 2023.

[26] Symeon Symeonidis, Dimitrios Effrosynidis, and Avi Arampatzis. A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. *Expert Systems with Applications*, 110:298–310, 2018.

[27] Nurulhuda Zainuddin and Ali Selamat. Sentiment analysis using support vector machine. In *2014 international conference on computer, communications, and control technology (I4CT)*, pages 333–337. IEEE, 2014.

[28] Yujia Zhai, Wei Song, Xianjun Liu, Lizhen Liu, and Xinlei Zhao. A chi-square statistics based feature selection method in text classification. In *2018 IEEE 9th International conference on software engineering and service science (ICSESS)*, pages 160–163. IEEE, 2018.