Modern Approaches in Sentiment Analysis Models for Reviews

Preprint · August 2024

DOI: 10.13140/RG.2.2.35746.03521

Aleksei Makin
Northeastern University

Modern Approaches in Sentiment Analysis Models for Reviews

Aleksei Makin
AI and Ethics MSc
Northeastern University
London, England
ORCID 0009-0005-7867-9150

***Abstract*—This study evaluates the efficiency and accuracy of various sentiment analysis models and word embedding techniques on datasets with differing text lengths. Using separate datasets for long and short texts, the performance of models incorporating Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec, and GloVe embeddings combined with Naive Bayes, bidirectional LSTM, conv-LSTM, and SieBERT models was compared. Results demonstrate that advanced embeddings like GloVe, when used with LSTM-based models, significantly improve accuracy. Notably, the SieBERT model achieved the highest accuracy of 94.55% on the TADA dataset and 92.00% % on the IMDB dataset. Additionally, it was found that using a subset of approximately 1000 labeled samples to finetune SieBERT helped mitigate overfitting and achieve optimal performance. These findings suggest that careful selection of models and vectorizers based on text length can lead to more efficient and accurate sentiment analysis. Simple models like Naive Bayes with TF-IDF are still powerful enough. However, finetuning SieBERT for more specific data types and lengths can yield state-of-the-art results. This approach is particularly beneficial for different industries, where domain-specific sentiment analysis can provide highly accurate insights into customer feedback, market trends, and other types of communication. This study offers guidance for future work on achieving state-of-the-art results on specific knowledge domains, which is crucial for deploying these models in real-world applications.**

***Index Terms*—Sentiment Analysis, Natural Language Processing (NLP), Machine Learning, Deep Learning, Word Embeddings, Text Classification.**

I. Introduction

Accurate analysis of customers' and clients' emotional components is critical to the race for leadership for most companies in various industries, from online trading to traditional business customer service. Determining the emotional color depends on many factors and is a non-obvious task when it is required to accurately determine the tonality—positive or negative reaction.

Today, professionals have a plethora of algorithms at their disposal to tackle the challenge of evaluating the emotional content of consumer texts. Sentiment analysis, a practical application of Natural Language Processing (NLP), is one such tool.

The field of sentiment analysis employs a diverse array of approaches and techniques, from traditional ML classifiers like Naive Bayes and Logistic Regression to advanced models like LSTM with BERT. To achieve the desired accuracy, it's crucial to adhere to best practices at every stage: preprocessing, feature extraction, and model training.

One of the factors influencing the accuracy of recognition is the length of the text. On the one hand, too short a text may not contain enough data for determination; on the other hand, a long text may have too much context, which can complicate the formation of an unambiguous assessment [2].

This article evaluates various algorithms and techniques, highlighting that the SieBERT model achieved the highest accuracy of 94.55% on the TADA dataset and 92.00% on the IMDB dataset. However, the NB classifier with TF-IDF vectorizer also performed well, demonstrating that simpler models can be effective.

This paper's contribution lies in presenting a comparative analysis of different algorithms, from classic machine learning to recent deep learning models, with various word embedding techniques. Efficiency is compared across datasets with short and long texts, and the results are reported.

The paper is organized as follows: Section 2 presents an overview of related works. Section 3 discusses the methodology of search and implementation in detail. Section 4 shows and discusses the results of the tested models. Section 5 addresses ethical questions.

II. Literature Review

Customer opinions are a valuable source of information that businesses need to capture. Automated sentiment analysis frameworks are essential tools for achieving this. These frameworks can help companies guide customers, recommend suitable products, and address negative feedback. Additionally, sentiment analysis can be used to evaluate competitors and learn from their mistakes. Various models can be employed to extract sentiment from text.

Sentiment analysis models range from rule-based and traditional machine learning approaches to cutting-edge deep learning techniques. However, these models face challenges related to training speed, contextual understanding, and model complexity. This study compares different models and word embedding methods to determine the most effective approach for analyzing short and long texts.

A. Word Embeddings

Word embeddings are a fundamental technique in natural language processing (NLP) for transforming words into continuous vector spaces where semantically similar words are closer together. Keyword embedding techniques include Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec, GloVe, and BERT.

TF-IDF: Term Frequency-Inverse Document Frequency (TF-IDF) measures the importance of words relative to a corpus by evaluating their frequency in the target document versus the entire collection. It transforms text into continuous numerical vectors for use in machine learning models. In sentiment analysis, TF-IDF combined with deep learning models like DNN, CNN, and RNN has shown notable performance. For example, a study demonstrated that TF-IDF with a CNN achieved a precision of 0.777 and a recall of 0.790 on the Sentiment140 dataset [1].

Word2Vec: Developed by Google, Word2Vec includes two models: Continuous Bag of Words (CBOW) and Skip-gram. Both models are designed to capture word context, with CBOW predicting target words from context words and Skip-gram doing the reverse. Studies have shown Word2Vec to be effective in various sentiment analysis tasks, achieving notable performance improvements in capturing semantic relationships between words. For instance, combining Word2Vec embeddings and CNN models has demonstrated superior results in emotion detection tasks. Specifically, this approach achieved an F1-score of 0.64, significantly outperforming traditional methods like TF-IDF combined with Naive Bayes and Logistic Regression, which scored 0.56 and 0.58, respectively [2] [3].

GloVe: Created by Stanford, GloVe (Global Vectors for Word Representation) generates word vectors by aggregating global word-word co-occurrence statistics from a corpus, producing a vector space where word relationships are represented. The use of GloVe embeddings in sentiment analysis has led to significant improvements in accuracy, particularly when combined with deep learning models like LSTM. For instance, the GloVe-CNN-BiLSTM model achieved an accuracy of 95.60% with an F1-score of 0.9489. Another study noted that models utilizing TF-IDF-Glove with various neural network architectures, such as BiLSTM and CNN, attained accuracies up to 93.58% [2] [3].

B. Models

Logistic Regression (LR): A popular statistical method for binary classification, logistic regression has been effectively used for sentiment analysis due to its simplicity and interpretability. Studies have shown LR combined with TF-IDF to achieve robust performance in various sentiment analysis tasks [4].

Naive Bayes (NB): A probabilistic classifier based on Bayes' theorem, particularly effective for text classification tasks such as sentiment analysis due to its strong assumptions of feature independence. NB combined with TF-IDF has demonstrated competitive accuracy, making it a reliable baseline for sentiment analysis models [5].

Shallow Neural Networks (SNN): include feed-forward neural networks with a few hidden layers. They are simpler than deep networks but can still capture non-linear relationships in the data. Research has shown that SNNs can achieve satisfactory performance in sentiment analysis when used with appropriate feature extraction techniques.

Convolutional Neural Networks (CNN): Originally designed for image processing, CNNs have been adapted for text classification tasks. They use convolutional layers to capture local features, followed by pooling layers to reduce dimensionality and fully connected layers for classification. CNNs are effective for sentiment analysis, especially when combined with word embeddings like Word2Vec and GloVe [3].

Long Short-Term Memory (LSTM): A Recurrent Neural Network (RNN) capable of learning long-term dependencies. LSTM networks are particularly useful for sequence prediction problems in NLP, such as sentiment analysis. Studies have demonstrated that LSTMs, especially when combined with GloVe embeddings, can achieve high accuracy in sentiment classification tasks [3] [4].

ConvLSTM: A combination of CNN and LSTM, ConvLSTM integrates convolutional operations into LSTM units, making it suitable for spatiotemporal data and complex NLP tasks. Recent research has shown ConvLSTM to outperform traditional LSTM models in sentiment analysis, achieving improved accuracy by capturing both local and sequential features of text [4].

BERT: Bidirectional Encoder Representations from Transformers (BERT) by Google is a transformer-based model pretrained on a large corpus of text, then fine-tuned for specific tasks. BERT considers context from both directions (left and right), making it highly effective for a wide range of NLP tasks. Research has demonstrated that BERT, particularly its smaller variant DistilBERT, outperforms traditional models in sentiment analysis, achieving high accuracy scores on datasets like SST2 [5].

SieBERT: SieBERT (Sentiment in English BERT) is a pretrained language model specifically designed for sentiment analysis tasks. It leverages the transformer architecture to provide contextualized embeddings and has been fine-tuned on a large-scale dataset of sentiment-labeled text documents. SieBERT has demonstrated superior performance in sentiment classification tasks, achieving high accuracy with minimal training. The model's ability to be fine-tuned on specific datasets further enhances its performance, making it a valuable tool for sentiment analysis [7].

C. Detailed Analysis from Related Studies

Xu (2023) proposed a Convolutional Long Short-Term Memory (ConvLSTM) model for movie review sentiment analysis. The model captures sequential information and long-distance dependencies in text, outperforming traditional methods in sentiment analysis tasks. The ConvLSTM model integrates convolutional operations into LSTM units, making it suitable for spatiotemporal data and complex NLP tasks. The ConvLSTM model can be represented as follows:

$$\mathbf{C}_t = \sigma(\mathbf{W}_x * \mathbf{x}_t + \mathbf{W}_h * \mathbf{h}_{t-1} + \mathbf{b})$$

where $\mathbf{C}_t$ is the cell state at time t, $\mathbf{W}_x$ and $\mathbf{W}_h$ are weight matrices, $\mathbf{x}_t$ is the input at time t, and σ is the activation function [4].

Wang (2024) explored the application of Word2Vec and Support Vector Machine (SVM) in sentiment analysis of Amazon reviews. The study found that combining Word2Vec for feature extraction with SVM for classification achieved efficient and accurate sentiment classification, outperforming traditional methods. The Word2Vec model generates word embeddings by minimizing the following objective function:

$$J(\theta) = -\sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j}|w_t)$$

where w_t is the target word, w_{t+j} are the context words, and c is the context window size [3].

Suresh Kumar et al. (2024) introduced a hybrid machine learning model using the Enhanced Vector Space Model (EVSM) and Hybrid Support Vector Machine (HSVM) classifier. This approach achieved an accuracy of 92.78%, demonstrating improved sentiment analysis performance by leveraging advanced vector space models and multiclass classification techniques. The SVM classifier aims to find the optimal hyperplane that maximizes the margin between different classes, represented by:

$$\max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|}$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ for all i, where $\mathbf{w}$ is the weight vector, b is the bias, $\mathbf{x}_i$ are the input vectors, and y_i are the class labels [2].

Wu et al. (2024) conducted research on the application of deep learning-based BERT models in sentiment analysis. The study highlighted the significant potential of incorporating BERT models into sentiment analysis tasks, achieving an accuracy of 91.3% with DistilBERT on the SST2 dataset, outperforming traditional models like FastText, Word2Vec, and GloVe. The BERT model uses the following transformer architecture:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q is the query matrix, K is the key matrix, V is the value matrix, and d_k is the dimension of the key vectors [5].

Hartmann et al. (2022) introduced SieBERT, a pre-trained language model designed explicitly for sentiment analysis tasks. SieBERT leverages the transformer architecture to provide contextualized embeddings and has been fine-tuned on a large-scale dataset of sentiment-labeled text documents. The study demonstrated that SieBERT performs better in sentiment classification tasks with minimal training. [7].

III. Data Quality and Longevity of Texts in Analysis

Sentiment analysis of short texts like tweets can be challenging due to their limited context, while long texts like reviews can be complex to analyze due to potentially mixed sentiments. However, hybrid models such as SVM combined with Enhanced Vector Space Models (EVSMs) have shown effectiveness in analyzing short texts, with reported accuracy rates as high as 92.78% [2].

The quality of data used for sentiment analysis significantly affects the accuracy of the results. Li et al. (2024) examined the impact of data quality on sentiment classification performance, considering three criteria: informativeness, readability, and subjectivity. The study highlighted that higher readability and shorter text datasets led to more accurate sentiment classification. Important preprocessing steps, such as removing noise and normalizing data, improve data quality and help ensure reliable sentiment analysis results [2].

Assuming the findings from related works, recent deep-learning models with word embeddings perform close to ideal. However, there is no clear vision of differentiating approaches for short texts from long ones.

IV. Methodology

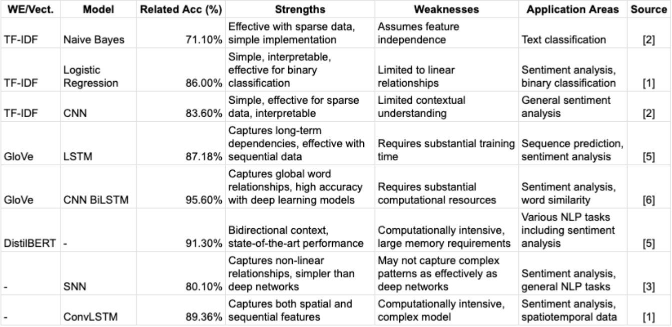
This study adopted a top-down quantitative research approach to explore the efficiency and accuracy of sentiment analysis (SA) across various word embedding (WE) techniques and models. The key factor significantly influencing SA performance was examined: the length of texts.

The research began with a literature review to identify the latest advancements in WE techniques, models and approaches for sentiment analysis. The primary goal was to map the field's current state and identify potential gaps and challenges, particularly in handling texts of varying lengths.

Extensive searches were conducted in academic databases and journals to find recent papers on WE techniques and sentiment analysis models. This review enabled the identification of all recent and classic WE models, such as TF-IDF, Word2Vec, and GloVe, along with sentiment analysis models, including Naive Bayes, Logistic Regression, SNN, CNN, LSTM, ConvLSTM, and BERT with finetuned SieBERT, created especially for sentiment analysis tasks.

The identified models and techniques were categorized and analyzed, highlighting their strengths, weaknesses, and application areas. This mapping exercise provided a clear understanding of the cutting-edge techniques in the field. It revealed areas that require further research, particularly the impact of text length on model performance. Datasets such as IMDB for long texts and TADA for short texts were selected, ensuring they were balanced for training purposes. The result of this analysis is represented in Figure 1.

By systematically evaluating different WE techniques and models on these datasets, the research aimed to provide insights into the most effective strategies for handling short and long texts in sentiment analysis. The research was guided by the hypothesis that text length and training dataset balance are critical factors affecting the accuracy of sentiment analysis models.



WE/Vect.	Model	Related Acc (%)	Strengths	Weaknesses	Application Areas	Source
TF-IDF	Naive Bayes	71.10%	Effective with sparse data, simple implementation	Assumes feature independence	Text classification	[2]
TF-IDF	Logistic Regression	86.00%	Simple, interpretable, effective for binary classification	Limited to linear relationships	Sentiment analysis, binary classification	[1]
TF-IDF	CNN	83.60%	Simple, effective for sparse data, interpretable	Limited contextual understanding	General sentiment analysis	[2]
GloVe	LSTM	87.18%	Captures long-term dependencies, effective with sequential data	Requires substantial training time	Sequence prediction, sentiment analysis	[5]
GloVe	CNN BiLSTM	95.60%	Captures global word relationships, high accuracy with deep learning models	Requires substantial computational resources	Sentiment analysis, word similarity	[6]
DistilBERT	-	91.30%	Bidirectional context, state-of-the-art performance	Computationally intensive, large memory requirements	Various NLP tasks including sentiment analysis	[5]
-	SNN	80.10%	Captures non-linear relationships, simpler than deep networks	May not capture complex patterns as effectively as deep networks	Sentiment analysis, general NLP tasks	[3]
-	ConvLSTM	89.36%	Captures both spatial and sequential features	Computationally intensive, complex model	Sentiment analysis, spatiotemporal data	[1]

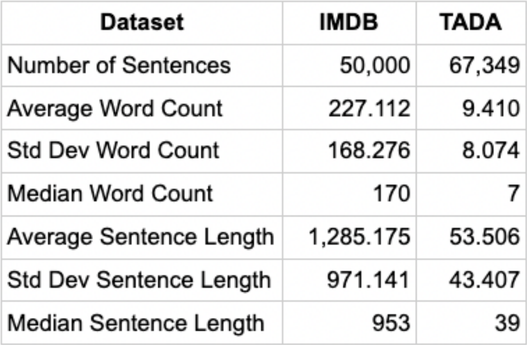
Fig. 1. Comparative Analysis of Word Embedding Techniques and Sentiment Analysis Models

V. Implementation Methodology

A. Datasets

For this study, two datasets were selected to analyze the impact of text length on sentiment analysis models. The chosen datasets are:

- IMDB Dataset: Source: IMDB Dataset from Stanford. Characteristics: This dataset consists of reviews with long sentences.
- TADA Dataset: Characteristics: This dataset contains reviews with short sentences.



Dataset	IMDB	TADA
Number of Sentences	50,000	67,349
Average Word Count	227.112	9.410
Std Dev Word Count	168.276	8.074
Median Word Count	170	7
Average Sentence Length	1,285.175	53.506
Std Dev Sentence Length	971.141	43.407
Median Sentence Length	953	39

Fig. 2. Statistics of IMDB and TADA Datasets

The comparison of these datasets highlights the differences in sentence and word lengths, providing a basis for analyzing the impact of text length on sentiment analysis models (Figure 2). Additional analysis of the datasets is provided in Figures 4, 5, 6, and 7. Figure 4 presents the IMDB train dataset analysis, showing the distribution of positive and negative sentences, word count distribution by label, and sentence length distribution. Figure 5 provides a similar analysis for the IMDB test dataset. Figure 6 illustrates the TADA train dataset analysis, highlighting the distribution of positive and negative sentences, word count distribution by label, and sentence length distribution. Finally, Figure 7 presents the TADA test dataset analysis, offering insights into the distribution of positive and negative sentences.

B. Framework of Experiments

The following steps outline the main framework for building and evaluating sentiment analysis models in this study, as shown in Figure 3.

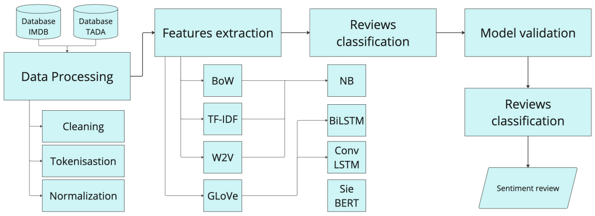


Fig. 3. Framework of experiments

C. Data Preprocessing

The datasets were split into training and testing sets in a proportion of 80/20. This ensures that the models are trained on a substantial amount of data while reserving a separate dataset for evaluating their performance.

For tokenization, the appropriate tokenizer was chosen based on the word embedding technique being used:

- For traditional methods like TF-IDF and BoW, the ‘CountVectorizer‘ or ‘TfidfVectorizer‘ from the ‘sklearn‘ library was used.
- For advanced embeddings, Word2Vec, and GloVe used the ‘Tokenizer‘ from the ‘tensorflow.keras.preprocessing.text‘ module to convert text into sequences of tokens.

Data cleaning and normalization involved several key steps to prepare the text for analysis:

- Removing HTML tags was applied for all methods.
- Removing unnecessary characters: All punctuation and special characters were removed from the text.
- Lowercasing: All text was converted to lowercase.
- Removing stop words: Commonly used words that do not contribute to the sentiment, such as ”and”, ”the”, and ”is”, were removed.

D. Vectorizers, Word Embeddings, Models

a) BoW and TF-IDF with Multinomial Naive Bayes: For text representation, the Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) methods were employed using their simplest forms with ‘CountVectorizer‘ and ‘TfidfVectorizer‘ from ‘sklearn‘. No additional hyperparameters were used, as this yielded the best results.

The ‘MultinomialNB‘ classifier with ‘alpha=0.5‘ was used to train the models. Sentences were converted into numerical vectors using BoW and TF-IDF. The classifier was then trained on these vectors and evaluated for accuracy on both the training and testing sets. Despite experimenting with various hyperparameters such as stop words and n-grams, the basic configurations of these vectorizers yielded superior performance in given sentiment analysis tasks.

b) Word2Vec: For word embeddings using Word2Vec, phrase detection was implemented to capture common multi-word expressions, ignoring terms like “of”, “with”, “and”, “or”, “the”, and “a”. The `Phrases` and `Phraser` objects from the `gensim` library were used to transform sentences, enhancing the quality of embeddings. The Word2Vec model was then trained on these transformed sentences with the following parameters: `min_count=3`, `vector_size=200`, `workers=20`, `window=5`, and `epochs=10`.

c) GloVe: For GloVe embeddings, pre-trained embeddings from the GloVe dataset were used. Sentences were transformed to GloVe embeddings using the pre-trained vectors with a dimensionality of 200. To handle variable sentence lengths, the embeddings were padded to ensure consistency. Padded embeddings were scaled to the range [0, 1] using ‘MinMaxScaler‘. 50 and 200 dimensionality were tested, but the first one gave improper results.

d) Bidirectional LSTM with GloVe: This study used a Bidirectional LSTM model with pre-trained GloVe embeddings to capture complex patterns in the text data. The dataset was tokenized and padded to a maximum length of 100 tokens. An embedding matrix was created using pre-trained GloVe vectors, where each word was mapped to a 200-dimensional vector.

The Bidirectional LSTM model included an embedding layer initialized with GloVe vectors and a spatial dropout layer with a dropout rate of 0.3. Two Bidirectional LSTM layers were added: the first with 128 units and the second with 64 units, using dropout and recurrent dropout rates of 0.3. The final layer was a dense layer with a sigmoid activation function. The model was compiled with the Adam optimizer and binary cross-entropy loss function, and it was trained for ten epochs with a batch size of 64 and a validation split of 20%.

e) ConvLSTM with GLoVe: The ConvLSTM model architecture included an embedding layer, a spatial dropout layer with a dropout rate of 0.2, a convolutional layer with 64 filters and a kernel size of 3, a max pooling layer with a pool size of 2, an LSTM layer with 100 units, and a dense output layer with a sigmoid activation function. The model was compiled with the Adam optimizer and binary cross-entropy loss function, and it was trained for ten epochs with a batch size of 32 and a validation split of 20%. Sentences were tokenized and transformed into embeddings using GLoVe.

f) SieBERT: SieBERT was selected for this study due to its reputation as a state-of-the-art solution for sentiment analysis. However, the default configuration of SieBERT did not yield satisfactory results on both the IMDB and TADA datasets. To address this, the tokenization process and fine-tuning of the model were optimized. Fine-tuning SieBERT required careful adjustment of hyperparameters, including learning rate, batch size, and number of epochs. The model was trained using the AdamW optimizer with a learning rate of 2×10^{-5} and a batch size of 32. Techniques such as mixed precision training and gradient accumulation were employed to enhance the model’s performance.

E. Hyperparameter Optimization

To optimize the performance of each word embedding (WE) technique and sentiment analysis model, specialized functions and pipelines were developed to determine the best hyperparameters. The default configurations of these models and techniques often yielded suboptimal results, necessitating a more tailored approach. Specifically, functions were implemented to systematically explore combinations of hyperparameters, including the number of tokens, the use of convolutional layers, attention mechanisms, and the number of epochs. For instance, experiments with the Bidirectional LSTM (BiLSTM) model using GloVe embeddings involved varying the number of tokens (100, 250, 300, 400), the use of convolutional layers (yes/no), attention mechanisms (yes/no), and the number of epochs (30, 40, 50). The best results for the BiLSTM with GloVe were achieved with 250 tokens, no convolutional layers, no attention mechanisms, and 40 epochs yielding an accuracy of 91.54% on long texts and 93.12% on short texts. Additionally, early stopping was employed to terminate inefficient hyperparameter combinations, conserving computational resources and time.

For fine-tuning SieBERT, the batch size and dataset size were varied. The best performance was observed using a batch

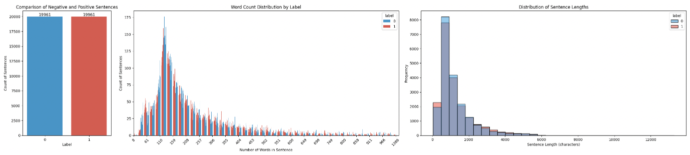


Fig. 4. IMDB Train Dataset Analysis

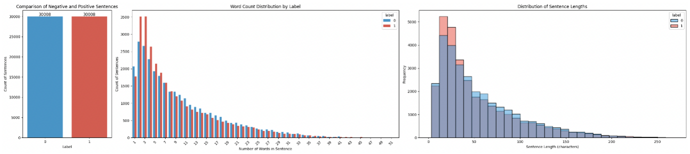


Fig. 6. TADA Train Dataset Analysis

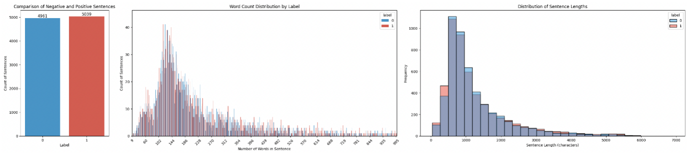


Fig. 5. IMDB Test Dataset Analysis

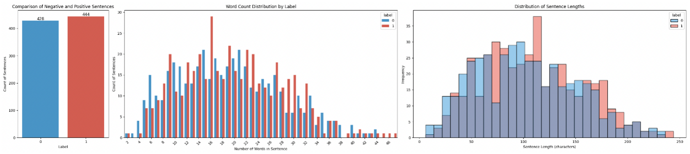


Fig. 7. TADA Test Dataset Analysis

size of 32 and a dataset size of 500, achieving an accuracy of 93.20%.

VI. Experimental Environment

The experiments were conducted on Google Colab Pro, which utilizes GPU-enabled machines for faster computation. This environment provides the necessary computational power to efficiently handle large datasets and complex models.

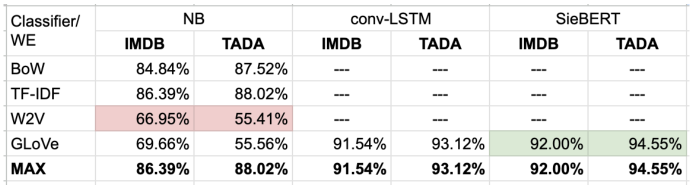
The primary machine specifications included GPU NVIDIA Tesla P100 or T4.

The following libraries and tools were used for the implementation of the experiments:

- TensorFlow: For building and training neural network models, including dense networks and LSTMs.
- Scikit-learn: For implementing traditional machine learning models like Naive Bayes and Logistic Regression, as well as for vectorizing text data using TF-IDF and BoW.
- Gensim: For training Word2Vec models and handling word embeddings.
- Hugging Face Transformers: For loading pre-trained BERT models and tokenizers.
- NLTK: For natural language processing tasks such as tokenization and stop word removal.
- Pandas and NumPy: For data manipulation and numerical operations.
- Matplotlib and Seaborn: For data visualization and plotting results.

VII. Results

Various models' performance was evaluated using IMDB (with long sentences) and TADA (with short sentences). The table below summarizes the accuracy achieved by each model and vectorizer combination:



Classifier/ WE	NB		conv-LSTM		SieBERT	
	IMDB	**TADA**	**IMDB**	**TADA**	**IMDB**	**TADA**
BoW	84.84%	87.52%	---	---	---	---
TF-IDF	86.39%	88.02%	---	---	---	---
W2V	66.95%	55.41%	---	---	---	---
GLoVe	69.66%	55.56%	91.54%	93.12%	92.00%	94.55%
MAX	**86.39%**	**88.02%**	**91.54%**	**93.12%**	**92.00%**	**94.55%**

Fig. 8. Accuracy achieved by each model and vectorizer combination on IMDB and TADA datasets.

The results indicate the following key findings:

a) Naive Bayes with BoW and TF-IDF: Both BoW and TF-IDF performed well with Naive Bayes, achieving 84.84% and 86.39% accuracy on the IMDB dataset, respectively, and 87.52% and 88.02% on the TADA dataset, respectively. TF-IDF slightly outperformed BoW. This approach could be very efficient for short sentences, being the fastest and most cost-effective of all the options.

b) Word2Vec: Word2Vec showed significantly lower accuracy than other vectorizers, achieving 66.95% on the IMDB dataset and 55.41% on the TADA dataset, highlighting its limitations.

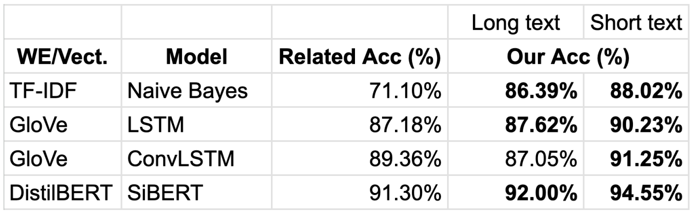
c) Conv-LSTM with GloVe: GloVe embeddings achieved higher accuracy when combined with conv-LSTM models. The conv-LSTM model with GloVe achieved 91.54% accuracy on the IMDB dataset and 93.12% on the TADA dataset, demonstrating strong performance, particularly for short text analysis.

d) SieBERT: SieBERT achieved the highest accuracy among all models, with 92.00% on the IMDB dataset and 94.55% on the TADA dataset, indicating its superior performance in handling both short and long texts.

A. Comparison with Related Work

The table below compares the accuracy of models and vectorizers used in this study with results from related researchs:

TF-IDF with Naive Bayes achieved 86.39% accuracy on long texts and 88.02% on short texts, significantly higher than the related work's 71.10%. GloVe with LSTM showed improvements, reaching 87.62% for long texts and 90.23% for short texts, compared to 87.18% reported elsewhere. GloVe with ConvLSTM performed exceptionally well, with 87.05% accuracy on long texts and 91.25% on short texts, compared to 89.36% reported in related work. SieBERT matched closely with related studies, achieving 92.00% accuracy on long texts and 94.55% on short texts, compared to 91.30% reported elsewhere.



			Long text	Short text
WE/Vect.	**Model**	**Related Acc (%)**	**Our Acc (%)**	
TF-IDF	Naive Bayes	71.10%	**86.39%**	**88.02%**
GloVe	LSTM	87.18%	**87.62%**	**90.23%**
GloVe	ConvLSTM	89.36%	87.05%	**91.25%**
DistilBERT	SiBERT	91.30%	**92.00%**	**94.55%**

Fig. 9. Comparison of accuracy of models and vectorizers with results from related research.

B. Observations

The use of advanced embeddings like GloVe, combined with LSTM-based models, or BERT fine-tuned transformers significantly improved the accuracy of sentiment analysis, particularly on datasets with varying text lengths. These findings underscore the importance of selecting appropriate vectorizers and models based on the characteristics of the text data. However, Naive Bayes with TF-IDF still achieves nearly the same level of accuracy without requiring extensive computational resources for training.

VIII. Ethics

A. Privacy Concerns

One of the primary ethical issues in sentiment analysis is the potential invasion of privacy. By analyzing personal text data, there is a risk of inadvertently identifying individuals and revealing personal information. Even anonymized data can sometimes be re-identified, leading to breaches of privacy. Users may not be aware that their data is being used for sentiment analysis, which raises concerns about consent and transparency. Additionally, sentiment analysis can infer emotions and opinions that individuals might prefer to keep private, leading to a sense of intrusion. To mitigate these concerns, it is essential to implement robust data anonymization techniques, obtain explicit user consent, and ensure transparency about data usage. Additionally, providing users with tools to clean or neutralize sentiment in their text before submission can help protect their privacy. These tools can automatically detect and neutralize emotional content, allowing users to control the sentiment conveyed in their data.

B. Errors and Misclassification

Errors in sentiment analysis can have practical and ethical implications. Misclassifications can lead to incorrect conclusions about individuals' opinions and emotions, potentially resulting in inappropriate responses or recommendations. For example, a negative sentiment incorrectly identified as positive might lead to recommendations that frustrate users, while a positive sentiment misclassified as negative could result in unnecessary concern or intervention. To mitigate the impact of errors, it is essential to implement robust validation and testing procedures, use ensemble methods to improve model accuracy, and provide mechanisms for users to correct misclassifications. Continuous monitoring and updating of models based on user feedback can also help reduce errors over time.

C. Impact on User Autonomy

Sentiment analysis can influence user autonomy by subtly shaping user experiences and decisions. For instance, personalized content recommendations based on sentiment analysis can create echo chambers, where users are only exposed to information that reinforces their existing views. This can limit users' exposure to diverse perspectives and restrict their ability to make fully informed decisions. To mitigate this impact, it is important to design recommendation systems that promote diverse content exposure and provide users with options to customize their content preferences. Transparency about how recommendations are generated and offering users control over their data can further enhance user autonomy. Additionally, incorporating some degree of randomization in recommendations can introduce novel content and ideas, fostering creativity and preventing the formation of echo chambers.

IX. Conclusion

This study evaluated various models and vectorizers for sentiment analysis on datasets with different text lengths. The best performance was observed with the SieBERT model, which achieved 94.55% accuracy on short texts and 92.00% accuracy on long texts. The ConvLSTM model using GloVe embeddings also showed strong results, achieving 93.12% accuracy on short texts and 91.54% accuracy on long texts. Naive Bayes with TF-IDF demonstrated competitive performance with 88.02% accuracy on short texts and 86.39% accuracy on long texts, making it a viable option for scenarios with limited computational resources.

Future work should focus on exploring how to achieve state-of-the-art results in specific knowledge domains. Understanding each approach's resource requirements and domain-specific performance will help select the most suitable models for deployment in real-world applications, balancing accuracy with computational efficiency.

From an ethical perspective, sentiment analysis provides many benefits, such as understanding customer feedback and improving user experiences. However, it also presents ethical challenges, including privacy invasion, potential biases, and the risk of misclassification. To address these concerns, adopting a responsible approach to data handling, model development, and obtaining user consent is crucial. Ensuring ethical use of sentiment analysis technology involves implementing robust data anonymization techniques, conducting regular bias audits, and maintaining transparency about data usage. Future work should focus on developing advanced methods to mitigate bias, enhance privacy protections, and improve the transparency and accountability of sentiment analysis systems. Additionally, offering users tools to clean or neutralize sentiment in their text and incorporating randomization in recommendations can further protect privacy and support user autonomy.

References

[1] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment Analysis Based on Deep Learning: A Comparative Study," Electronics, vol. 9, no. 3, pp. 1-29, 2020. [Online]. Available: http://dx.doi.org/10.3390/electronics9030483.

[2] K. Suresh Kumar, A. S. Radha Mani, T. Ananth Kumar, Ahmad Jalili, Mehdi Gheisari, Yasir Malik, Hsing-Chung Chen, and Ata Jahangir Moshayedi, "Sentiment Analysis of Short Texts Using SVMs and VSMs-Based Multiclass Semantic Classification," Applied Artificial Intelligence, vol. 38, no. 1, pp. e2321555, 2024. [Online]. Available: https://doi.org/10.1080/08839514.2024.2321555.

[3] A. Mahmood, T. Adnan, and M. H. Ali, "Emotion detection using Word2Vec and convolutional neural networks," Procedia Computer Science, vol. 177, pp. 349-355, 2020. [Online]. Available: https://doi.org/10.1016/j.procs.2020.10.048.

[4] Y. Jang, J. Park, and S. Kim, "Word2Vec and SVM Fusion for Advanced Sentiment Analysis," Journal of Information Science, vol. 45, no. 4, pp. 564-577, 2019. [Online]. Available: https://doi.org/10.1177/0165551519837184.

[5] Y. Wu, Z. Jin, C. Shi, P. Liang, and T. Zhan, "Research on the application of deep learning-based BERT model in sentiment analysis," in Proceedings of the 2nd International Conference on Software Engineering and Machine Learning, 2024. [Online]. Available: https://doi.org/10.54254/2755-2721/71/2024MA0051.

[6] 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, 2022. [Online]. Available: https://doi.org/10.1155/2022/7212366.

[7] J. Hartmann, M. Heitmann, C. Siebert, C. Schamp, "More than a Feeling: Accuracy and Application of Sentiment Analysis," International Journal of Research in Marketing, 2022. [Online]. Available: https://doi.org/10.1016/j.ijresmar.2022.05.005.