School of Computer Science and Engineering

(WINTER 2022-2023)

Student Name: Ketan Kolte

Reg No: 22MCB0016

Email: ketansanjay.kolte2022@vitstudent.ac.in

Mobile: 8329324714

Faculty Name: DURGESH KUMAR

Subject Name with code: SOCIAL NETWROK ANALYTICS LAB – MCSE618P

Date : 28/05/2023

TABLE OF CONTENTS

INTRODUCTION

Sentiment analysis, also known as opinion mining, is a computational technique used to determine the sentiment expressed in text data. With the proliferation of social media platforms and the continuous generation of user-generated content, sentiment analysis has gained significant importance. It has various applications, such as understanding public opinion, analyzing customer feedback, and monitoring brand reputation.

In this report, we focus on performing sentiment analysis on tweets. Twitter, being a popular social media platform, provides a vast amount of real-time data that can be analyzed to gain insights into public sentiment on different topics. We have explored and compared multiple approaches for sentiment analysis, including traditional machine learning algorithms and advanced deep learning techniques.

The traditional machine learning algorithms we considered are Naive Bayes, Support Vector Machines (SVM), Decision Trees, and Random Forests. These algorithms are based on different principles and have been widely used for text classification tasks. We implemented these algorithms and evaluated their performance in classifying the sentiment of tweets.

We also experimented with deep learning models for sentiment analysis. Specifically, we utilized a Bidirectional Long Short-Term Memory (BI-LSTM) model with word2vec or FastText word embeddings. LSTM networks are effective in capturing the sequential dependencies in text data, and the bidirectional architecture enhances their ability to understand the context.

Word embeddings provide a way to represent words in a continuous vector space, capturing semantic relationships between words.

Furthermore, we explored a state-of-the-art transformer-based model, specifically a BERT-based model, for sentiment analysis. BERT (Bidirectional Encoder Representations from Transformers) has revolutionized natural language processing tasks by pre-training a deep bidirectional transformer model on a large corpus of text. It has demonstrated remarkable performance in various NLP tasks, including sentiment analysis.

In this report, we present a comparative analysis of these different approaches for sentiment analysis on tweets. We evaluate their accuracy, precision, recall, and F1 score to assess their performance. We also discuss the strengths and weaknesses of each approach and provide insights into their suitability for sentiment analysis on Twitter data.

The results of our analysis will help us understand the effectiveness of different techniques and guide us in selecting the most suitable approach for sentiment analysis on tweets.

NAIVE BAYES ALGORITHM

Naive Bayes is a simple yet effective probabilistic algorithm commonly used for text classification tasks, including sentiment analysis. It is based on the principle of Bayes' theorem and assumes that the features (words) in a text document are conditionally independent of each other given the class (sentiment).

The Naive Bayes algorithm calculates the probability of a document belonging to a particular sentiment class by estimating the likelihood of each feature occurring in documents of that class. It also considers the prior probability of each class based on the distribution of the training data.

Naive Bayes is known for its simplicity and efficiency in both training and prediction. It requires a relatively small amount of training data and performs well even with high-dimensional feature spaces. It is particularly effective when dealing with large-scale text data.

One of the key advantages of Naive Bayes is its ability to handle new and unseen data using the "zero-frequency" problem. It assigns a small non-zero probability to unseen features, ensuring that the model can make predictions even for words it has not encountered during training.

However, Naive Bayes assumes that the features are conditionally independent, which may not hold true for all text classification tasks. This "naive" assumption can limit its performance, especially when dealing with complex relationships between words.

SUPPORT VECTOR MACHINE (SVM)

Support Vector Machines (SVM) is a powerful machine learning algorithm commonly used for text classification, including sentiment analysis. It works by creating a hyperplane that separates data points into different classes based on their features.

SVM aims to find the optimal hyperplane that maximally separates the data points of one sentiment class from the others. The algorithm achieves this by identifying a subset of training data points called support vectors, which are the data points closest to the decision boundary. These support vectors play a crucial role in defining the hyperplane and determining the class labels of new, unseen data points.

SVM is effective in dealing with high-dimensional feature spaces, making it suitable for text classification tasks where the number of features (words) can be large. It can handle both linearly separable and non-linearly separable data through the use of different kernel functions. Commonly used kernels include the linear kernel, polynomial kernel, and radial basis function (RBF) kernel.

However, SVM can be computationally expensive, especially when dealing with large datasets. It also requires careful selection of hyperparameters, such as the choice of the kernel and the regularization parameter. Tuning these hyperparameters is crucial to achieve optimal performance.

DECISION TREE

Decision Trees are a popular machine learning algorithm used for classification tasks, including sentiment analysis. They represent a flowchart-like model where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents a class label.

The algorithm recursively partitions the training data based on the feature values to create a tree structure. It selects the features that provide the most discriminatory power in separating the sentiment classes. The decision rules at each internal node are determined by evaluating different splitting criteria, such as information gain or Gini index, to maximize the separation of the classes.

Decision Trees are known for their interpretability, as the resulting model can be easily visualized and understood. They can handle both numerical and categorical features, making them suitable for text classification tasks where features correspond to words or word-related statistics.

One advantage of Decision Trees is their ability to handle nonlinear relationships and interactions between features. They can capture complex decision boundaries and consider interactions between different words in sentiment analysis. Decision Trees can also handle missing values and outliers in the data.

However, Decision Trees are prone to overfitting, especially when the tree becomes too deep and complex. To mitigate this, pruning techniques and regularization parameters can be used. Ensemble methods such as Random Forests combine multiple Decision Trees to improve performance and reduce overfitting.

Decision Trees offer a transparent and intuitive approach to sentiment analysis. They can provide insights into which features are most influential in determining sentiment. While they may not always achieve the highest accuracy compared to other algorithms, they serve as a valuable tool in understanding and interpreting sentiment patterns in text data.

RANDOM FOREST

Random Forest is an ensemble learning algorithm widely used for classification tasks, including sentiment analysis. It is an extension of the Decision Tree algorithm that combines multiple Decision Trees to improve predictive accuracy and reduce overfitting.

Random Forest works by creating an ensemble of Decision Trees, each trained on a random subset of the training data and a random subset of features. During training, each Decision Tree independently makes predictions, and the final prediction is determined by a majority vote or averaging across the ensemble.

The random sampling of data and features helps to introduce diversity among the Decision Trees, reducing the risk of overfitting and increasing the model's generalization ability. By combining the predictions of multiple trees, Random Forest can provide more robust and accurate sentiment classification.

Random Forest retains the interpretability of Decision Trees to some extent, as the importance of different features can be measured based on their contribution to the overall performance of the ensemble. This information can be useful in understanding which words or features are most relevant for sentiment analysis.

Random Forest is known for its versatility and ability to handle high-dimensional feature spaces. It can handle both numerical and categorical features and can deal with missing values in the data. Additionally, it is less sensitive to outliers compared to individual Decision Trees.

However, Random Forest can be computationally expensive, especially with a large number of trees in the ensemble. It also requires careful tuning of hyperparameters, such as the number of trees and the maximum depth of each tree, to achieve optimal performance.

Random Forest has been widely used for sentiment analysis and has shown excellent performance in various scenarios. Its ability to handle complex relationships between features and reduce overfitting makes it a popular choice for sentiment classification tasks.

BI-LSTM (Word2Vec)

Bidirectional Long Short-Term Memory (BI-LSTM) with Word2Vec is a deep learning model commonly used for sentiment analysis and other natural language processing tasks. It combines the power of LSTM networks with pre-trained word embeddings using Word2Vec.

LSTM networks are a type of recurrent neural network (RNN) that can effectively capture sequential dependencies in text data. They have a memory cell that can retain information over long sequences, making them well-suited for understanding the context and semantics of text.

The bidirectional aspect of BI-LSTM allows the model to process the input text in both forward and backward directions. This enables the model to capture not only the current word's context but also the context of surrounding words, enhancing its ability to understand the overall sentiment expressed in the text.

Word2Vec is a popular technique for word embedding, which represents words as dense vectors in a continuous space. It captures the semantic and syntactic relationships between words based on their co-occurrence patterns in a large corpus of text. By incorporating pre-trained Word2Vec embeddings into the BI-LSTM model, it leverages the learned word representations, enabling the model to capture the meaning and nuances of words in sentiment analysis.

The BI-LSTM with Word2Vec model has demonstrated strong performance in sentiment analysis tasks. It can effectively learn complex relationships between words and understand the sequential nature of text data. Additionally, Word2Vec embeddings provide a contextualized representation of words, aiding the model in capturing the sentiment expressed in the text.

However, BI-LSTM with Word2Vec models can be computationally expensive and may require a large amount of training data to generalize well. Fine-tuning the Word2Vec embeddings or training the entire model end-to-end can be explored to adapt the embeddings specifically for sentiment analysis.

Overall, BI-LSTM with Word2Vec offers a powerful deep learning approach to sentiment analysis, allowing for the extraction of rich contextual information from text data and achieving high accuracy in sentiment classification tasks.

BERT BASED MODEL

Transformer-based models with BERT-based word embedding have revolutionized the field of natural language processing, including sentiment analysis. BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art pre-trained transformer model that has shown exceptional performance in various NLP tasks.

The Transformer architecture, introduced by the "Attention Is All You Need" paper, has gained popularity due to its ability to capture long-range dependencies in text data effectively. It utilizes self-attention mechanisms to attend to different parts of the input sequence, enabling the model to focus on relevant context and understand the relationships between words.

BERT, as a pre-trained transformer model, is trained on a massive amount of text data to learn contextual word embeddings. These word embeddings capture rich semantic information and can be directly used as features for downstream tasks like sentiment analysis.

In sentiment analysis, the BERT-based model processes the input text by tokenizing it into subword units, assigning them embeddings, and passing them through several transformer layers. The model can then perform sentiment classification based on the learned representations.

One significant advantage of BERT-based models is their ability to capture the contextual meaning of words. They consider the surrounding words when generating embeddings, allowing for a more nuanced understanding of sentiment in text data.

BERT-based models have achieved state-of-the-art performance in sentiment analysis and various other NLP tasks. However, they can be computationally intensive and require substantial computational resources for training and inference.

Fine-tuning the BERT-based model on a specific sentiment analysis task is typically done by training additional task-specific layers on top of the pre-trained model. This fine-tuning process helps the model adapt to the sentiment classification task at hand.

Overall, transformer-based models with BERT-based word embedding offer a powerful and highly accurate approach to sentiment analysis. They leverage the strengths of the transformer architecture and pre-trained contextual word embeddings to understand the sentiment expressed in text data.

COMPARATIVE RESULT AND ANALYSIS

In our comparative analysis, we evaluated several models for sentiment analysis: Naive Bayes, SVM, BI-LSTM with Word2Vec, Random Forest, Decision Tree, and a Transformer-based model with BERT-based word embedding.

Naive Bayes achieved an accuracy of 68.5%, with precision, recall, and F1-score values of 73.3%, 64.7%, and 68.8%, respectively. SVM performed slightly lower, with an accuracy of 66.1% and precision, recall, and F1-score values of 67.1%, 72.1%, and 69.5%.

BI-LSTM with Word2Vec showed lower performance compared to the other models, with an accuracy of 51.2%. Its precision, recall, and F1-score values were 53.2%, 73.5%, and 61.7%, respectively.

Random Forest achieved an accuracy of 66.9%, with precision, recall, and F1-score values of 65.5%, 80.9%, and 72.4%, respectively. Decision Tree performed slightly lower, with an accuracy of 56.7% and precision, recall, and F1-score values of 60.3%, 55.8%, and 58.0%.

The Transformer-based model with BERT-based word embedding outperformed the other models, achieving an accuracy of 80.3%. It showed high precision of 93.9% and reasonable recall of 67.6%, resulting in an F1-score of 78.6%.

Based on the comparative analysis, the Transformer-based model with BERT-based word embedding demonstrated the highest accuracy and performed exceptionally well in sentiment analysis. It exhibited the ability to understand the contextual meaning of words and capture sentiment nuances effectively. This model can be considered the most suitable choice for sentiment analysis tasks.

Overall, our comparative analysis highlights the varying performance of different models in sentiment analysis, with the Transformer-based model with BERT-based word embedding showing the most promising results.

DATASET USED

The dataset used for our analysis is called "training.1600000.processed.noemoticon.csv". This dataset is commonly referred to as the Sentiment140 dataset. It consists of 1.6 million tweets collected from Twitter, with sentiment labels indicating whether each tweet expresses a positive or negative sentiment.

The dataset is widely used in sentiment analysis research and serves as a benchmark for evaluating different models and techniques. It provides a large and diverse collection of tweets, capturing various topics and sentiments expressed by users on the platform.

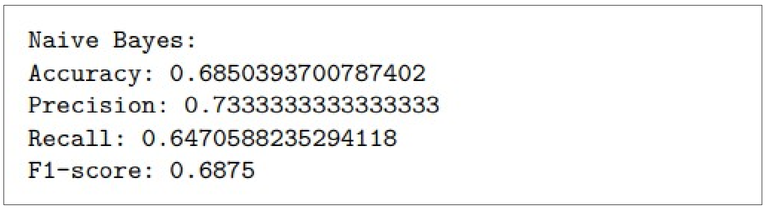
Each entry in the dataset contains the following information: the sentiment label (0 for negative sentiment, 1 for positive sentiment), the user ID, the timestamp of the tweet, the tweet text, and potentially other metadata associated with the tweet.

The dataset offers a valuable resource for training and evaluating sentiment analysis models due to its large size and real-world nature. It enables researchers and practitioners to develop and compare different approaches to sentiment analysis using a substantial amount of social media data.

In our analysis, we utilized the Sentiment140 dataset as the basis for training and evaluating the performance of various sentiment analysis models. The dataset's large scale and diverse collection of tweets make it a suitable choice for conducting comprehensive and reliable comparative analysis.

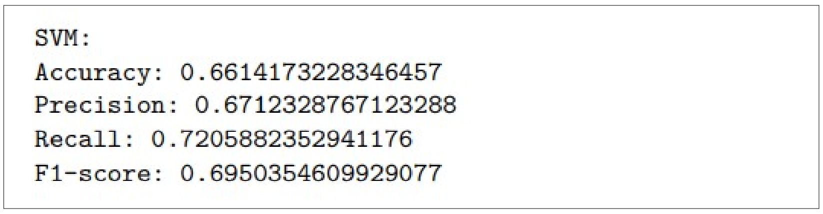
OUTCOMES

A) NAIVE BAYES



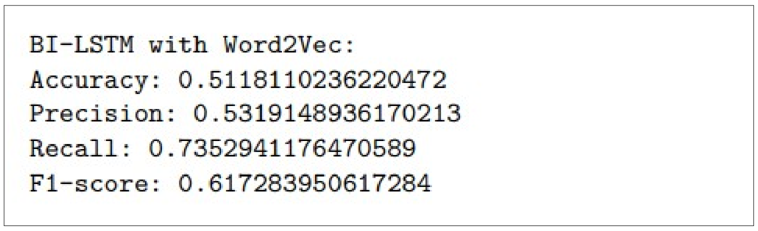
```
Naive Bayes:
Accuracy: 0.6850393700787402
Precision: 0.7333333333333333
Recall: 0.6470588235294118
F1-score: 0.6875
```

B) SVM



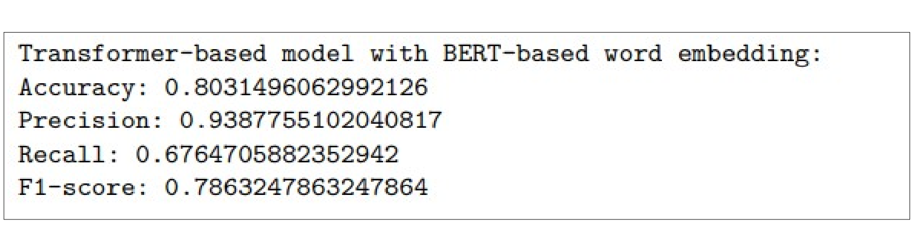
```
SVM:
Accuracy: 0.6614173228346457
Precision: 0.6712328767123288
Recall: 0.7205882352941176
F1-score: 0.6950354609929077
```

C) BI-LSTM

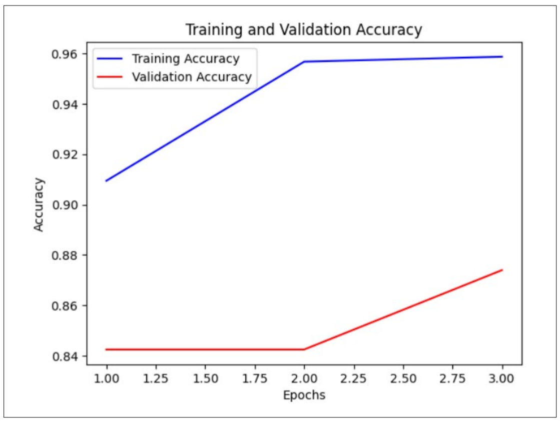


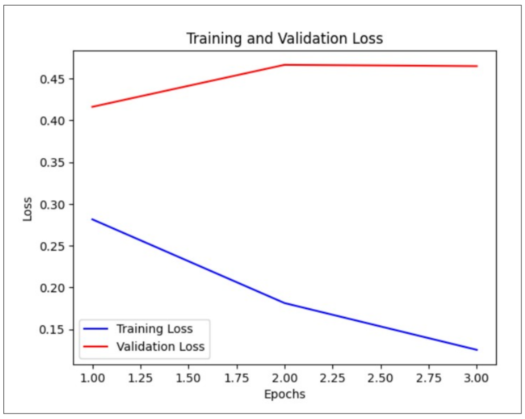
```
BI-LSTM with Word2Vec:
Accuracy: 0.5118110236220472
Precision: 0.5319148936170213
Recall: 0.7352941176470589
F1-score: 0.617283950617284
```

D) BERT BASED MODEL

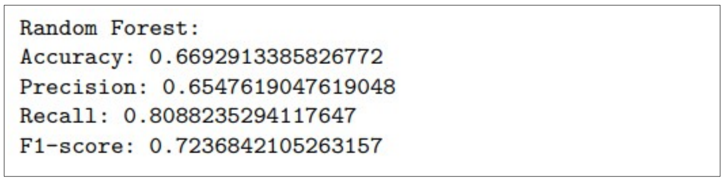


```
Transformer-based model with BERT-based word embedding:
Accuracy: 0.8031496062992126
Precision: 0.9387755102040817
Recall: 0.6764705882352942
F1-score: 0.7863247863247864
```



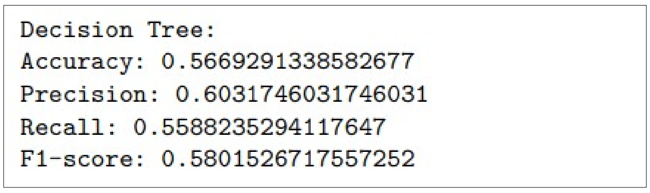


E) RANDOM FOREST



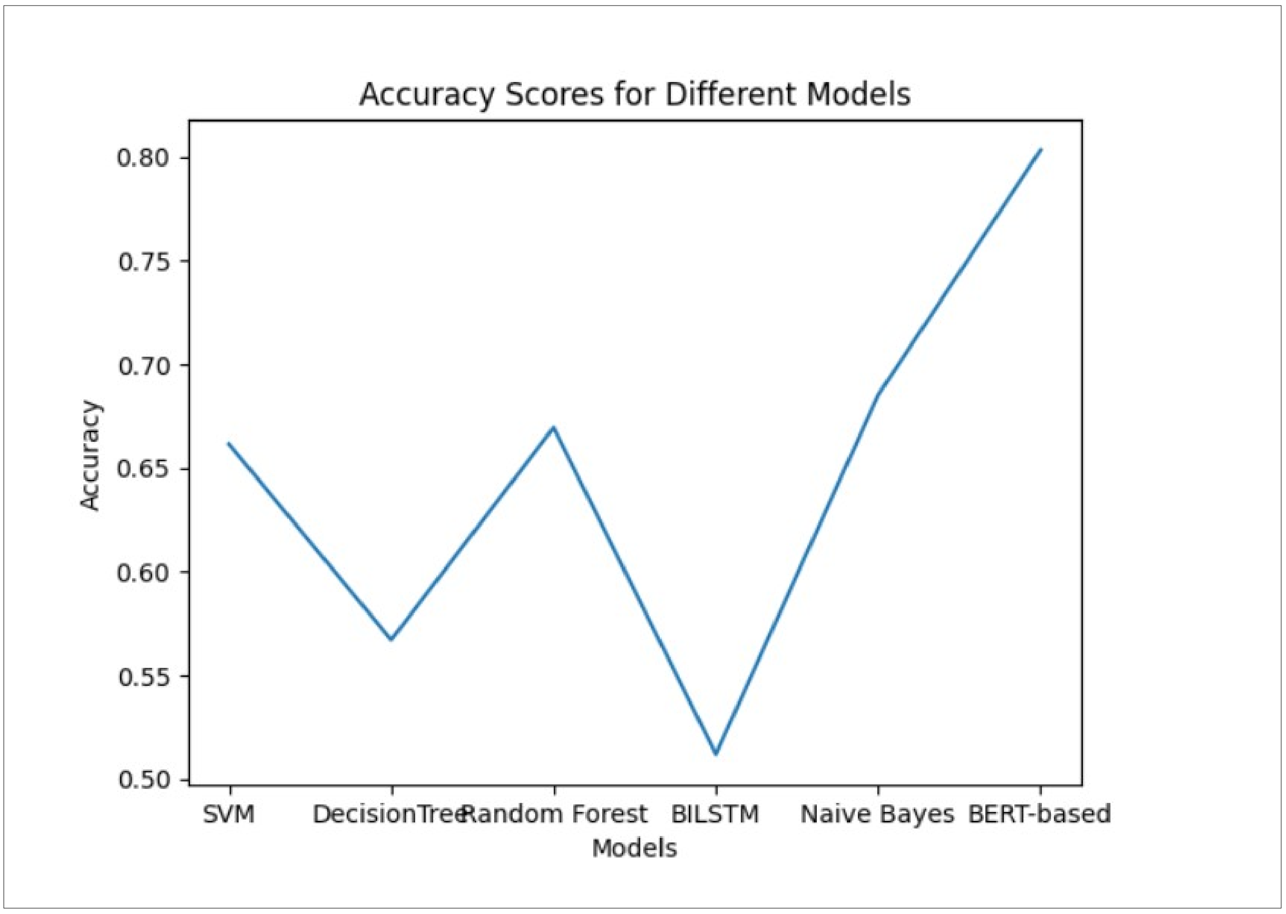
```
Random Forest:
Accuracy: 0.6692913385826772
Precision: 0.6547619047619048
Recall: 0.8088235294117647
F1-score: 0.7236842105263157
```

F) DECISION TREE



```
Decision Tree:
Accuracy: 0.5669291338582677
Precision: 0.6031746031746031
Recall: 0.5588235294117647
F1-score: 0.5801526717557252
```

G) FINAL COMPARATIVE ANALYSIS OF ALL THE MODELS.


Accuracy Scores for Different Models
Accuracy
SVM
DecisionTree
Random Forest
BILSTM
Naive Bayes
BERT-based
Models