Sentiment Analysis on Twitter Data

Shushant Ghosh
Khoury College of Computer Sciences
Northeastern University Boston MA 02115
ghosh.shu@northeastern.edu

Samuel Ghebreyesus
Khoury College of Computer Sciences
Northeastern University Boston MA 02115
ghebreyesus.s@northeastern.edu

Arya Dhorajiya
Khoury College of Computer Sciences
Northeastern University Boston MA 02115
dhorajiya.a@northeastern.ed

Abstract

With social media becoming the most important medium for communication and expression in the current times, the need for tools to help manage the sudden abundance of Sentiment Data has become very prominent. Detecting sentiment of tweets and their extent has a huge number of use cases ranging from customer support management to detecting suicidal tendencies. With our Project , we aim at using Supervised and Unsupervised Learning methods to make the detection of Sentiments in tweets even more effective and efficient.

1. Introduction

In this project, our group aimed to implement a model for performing sentiment analysis on Twitter data using Kaggle's Sentiment140 dataset, consisting of 1.6 million tweets and their corresponding polarity(encoded as 0, 2 and 4 for positive, negative and neutral respectively). We wanted to be able to take an unseen tweet and output the sentiment as either positive, negative or neutral. We tested a lot of supervised and unsupervised learning methods. For supervised methods, we tried probabilistic classifiers such as Naive Bayes as well as using Support Vector Machines and Logistic Regression. For unsupervised methods, we tried dimensionality reduction methods like T-SNE, clustering algorithms like K-means and lexical-based approaches such as Vader and Textblob. With each of these models, we compared performance across the metrics of precision, recall and F1 score.

Exploratory Data Analysis

Before putting the tweets into any models, we had to do some preprocessing and exploratory data analysis. First, we cleaned the data by dropping every column except for the text and polarity columns. Next, we checked to see if there were any null values in our dataset that needed to be taken care of and found none. Furthermore, we examined the distribution of the target variable in our dataset and found that the dataset was perfectly balanced and thus, things like upsampling minority classes would not need to be done. In terms of preprocessing, the main things we did were remove stopwords, punctuation, repeating characters, URLs, numbers, and stemming/lemmatization. Afterwards, we created a histogram to show the distribution of tweet length for each of the positive/negative sentiment classes to see if there was a significant difference. From the histogram, we found that shorter tweets were more likely to be positive and longer tweets more likely to be negative but the difference was pretty miniscule/insignificant. Lastly, we created word clouds and bar graphs to display the most common words in each class that weren't stopwords. Overall, we found there was a lot of overlap in most common words along each class and thus this metric was probably not going to be super helpful in helping us determine sentiment.

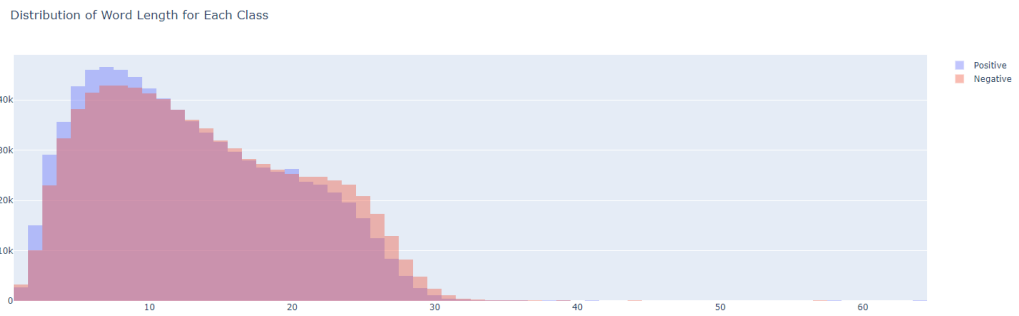


Fig 1. Distribution of word lengths for Positive and Negative Tweets

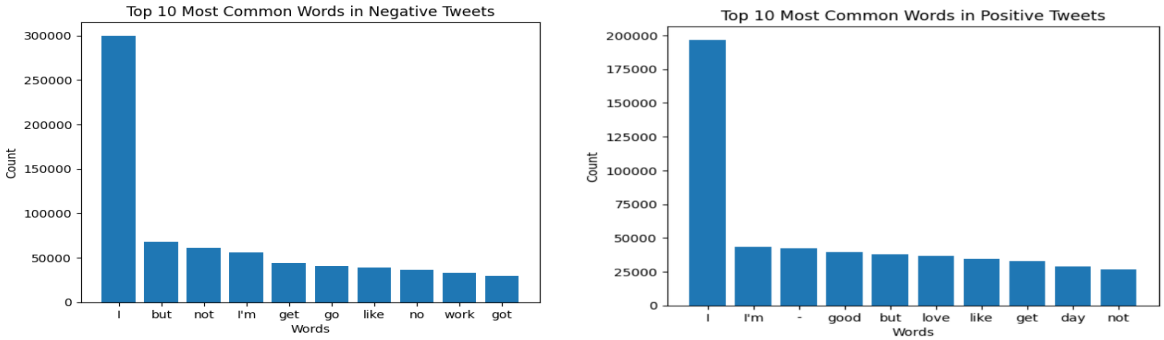


Fig 2. Most Common words in the Negative and Positive sentiment tweets

2. Methods

This section will provide a brief overview of data preprocessing techniques, and we will apply both supervised and unsupervised approaches to the Sentiment140 dataset. The supervised approach will involve training a machine learning model using a portion of the labeled dataset as training data, followed by evaluating the model's performance on the remaining data. On the other hand, the unsupervised approach will utilize clustering and topic modeling techniques to identify patterns and sentiments in the data, without relying on labeled examples. By utilizing both approaches, we aim to gain a comprehensive understanding of the sentiment present in the Sentiment140 dataset, as well as the relative strengths and limitations of each method.

2.1 Data Preprocessing

The Sentiment140 dataset is a collection of 1.6 million tweets with a sentiment label of either positive (1) or negative (0). To preprocess the data for sentiment analysis, steps given below were typically followed:

1. Cleaning: The first step is to remove any unnecessary information from the text data, such as URLs, hashtags, mentions, and special characters. We had to implement this step as it reduces the dimensionality of the data and removes any noise that could affect the performance of the model.
2. Tokenization: Our next step was to break down the text into individual words or tokens. Tokenization is essential for further processing the text data, such as removing stop words and stemming.
3. Stop word removal: Stop words are words that occur frequently in a language, such as "the," "and," and "a." These words do not carry much meaning so we removed them using Stop word libraries.
4. Stemming: Stemming is the process of reducing words to their root form. This helps to group similar words together and reduces the dimensionality of the data. For instance, "running," "runner," and "run" are stemmed to "run."

5. Vectorization: Once the text was cleaned and tokenized,we converted it into a numerical representation that can be fed into a machine learning model. We used the Term Frequency-Inverse Document Frequency (TF-IDF) technique which assigns a weight to each word in the text based on its frequency of occurrence across all documents in the corpus.
6. Splitting: Finally, we splitted the preprocessed data into training and testing sets to evaluate the performance of the machine learning model.

By following these steps, the given data corpus had been preprocessed to reduce Training time and improve Performance on given Machine Learning Algorithms.

2.2 Supervised Models

Supervised learning models require labeled data for training. For our project, we have selected three supervised models: Naive Bayes, Support Vector Machines (SVM), and Logistic Regression (LR).

2.2.1 Naive Bayes

Naive Bayes is a probabilistic model that works on the principle of Bayes' theorem. It assumes that the presence of a particular feature in a class is independent of the presence of any other feature. The Naive Bayes classifier is simple and efficient, making it a popular choice for sentiment analysis. The evaluation of Naive Bayes will be done using Precision, recall, and F1 score.

2.2.2 Support Vector Machines

Support Vector Machines is a powerful model for classification tasks. It works by finding the hyperplane that maximally separates the classes. SVM can handle high-dimensional data and is often used in text classification tasks. The evaluation of SVM will be done using Precision, Recall, and F1 score.

2.2.3 Logistic Regression

Logistic Regression is a probabilistic model that works by modeling the probability of a certain class. It is a popular model for binary classification tasks. The evaluation of logistic regression will be done using Precision, Recall, and F1 score.

2.3 Unsupervised Models

Unsupervised learning models do not require labeled data for training. For the given project, we have selected three unsupervised models: t-SNE, K-means, and Lexicon-based approach.

2.3.1 t-SNE

t-SNE is a dimensionality reduction technique that is commonly used for visualizing high-dimensional data. It works by minimizing the divergence between two probability distributions: a Gaussian distribution that measures pairwise similarities in the high-dimensional space and a t-distribution that measures pairwise similarities in the low-dimensional space. The evaluation of t-SNE will be done using visualization of the clusters.

2.3.2 K-means

K-means is a clustering algorithm that works by partitioning the data into k clusters. It works by minimizing the sum of squared distances between the data points and their cluster centroids. The evaluation of k-means will be done using predicting the clusters and comparing it with the original given labels.

2.3.3 Lexicon-based approach

The lexicon-based approach works by using a sentiment lexicon, which is a collection of words with their corresponding sentiment scores. The sentiment score of a text is calculated by aggregating the scores of the words in the text. The evaluation of the lexicon-based approach will be done using Precision, Recall, and F1 score.

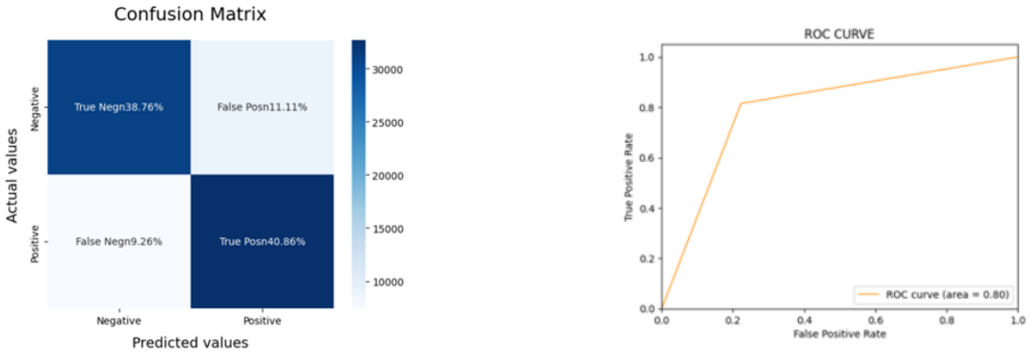
2.4 Training

The paragraph describes the process of supervised learning, where models are trained on 95% of the original dataset consisting of 1.52 million data points. The Sentiment140 dataset is used, where tweets are labeled as positive or negative (1 or 0), and the features are converted into TF-IDF matrices. The models are then tested on a separate testing set to determine their performance metrics, using Binary Cross Entropy as the loss function to calculate the error and other performance metrics. The training time for each model was around 100 seconds.

The paragraph describes the use of unsupervised learning on the Sentiment140 dataset to assess the performance of clustering algorithms and lexicon-based approaches. The data was initially visualized using t-SNE to obtain a 2-D distribution. Subsequently, the K-means algorithm was applied to the training data to predict clusters, but it resulted in poor performance. As a result, VADER and TextBlob libraries, designed for sentiment analysis, were used to calculate the Sentiment Score. Documents with a positive score were assigned the label '1', while those with a negative score were assigned the label '0'. Clustering approach took around 1200 seconds, while utilizing VADER and TextBlob Libraries took 113 seconds.

3. Results

3.1 Supervised Sentiment Analysis



(a) Naïve Bayes

(a) ROC Curve (Naïve Bayes)

Fig 3 : Confusion Matrices and ROC curve for Supervised method..

Models	Precision	Recall	F1-Score
Naïve Bayes	80%	81%	80%
SVM	81%	79%	80%
Logistic Regression	82%	80%	81%

Table 1 : Performance Metrics for Supervised Sentiment Analysis

The models were trained on a corpus and their performance was evaluated on a separate testing corpus consisting of 80,000 data points. Despite its simplicity, Logistic Regression outperformed Naive Bayes and SVM, achieving an F1-score of around 81%. Although Naive Bayes and SVM are considered state-of-the-art models for supervised

analysis on textual data, they achieved slightly lower F1-scores of 80%. The ROC curves for all models were similar, indicating that they are suitable for sentiment analysis.

3.2 Unsupervised Sentiment Analysis

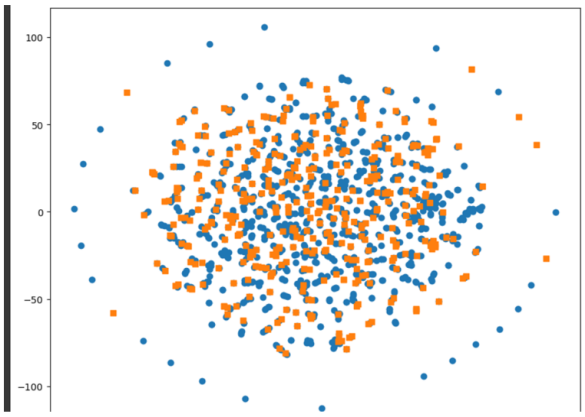


Fig 4: Visualizing data spread in 2-D, after applying t-SNE to TF-IDF Matrix of documents

The figure provided depicts that the application of t-SNE to our TF-IDF matrix did not result in clearly separated data points, suggesting that unsupervised algorithms may not perform well. However, this hypothesis needs to be further tested to confirm its validity.

Hypothesis: The performance of dimensionality reduction methods on text data may not be optimal.

Models	Precision	Recall	F1-Score
K-Means	56%	53%	53%
K-Means + PCA	40%	49%	49%
K-Means + t-SNE	49%	49%	49%
Hierarchical Cl. + PCA	49%	49%	49%
Hierarchical C. + t-SNE	41%	49%	48%

Table 2: Performance Metrics for Unsupervised Sentiment Analysis

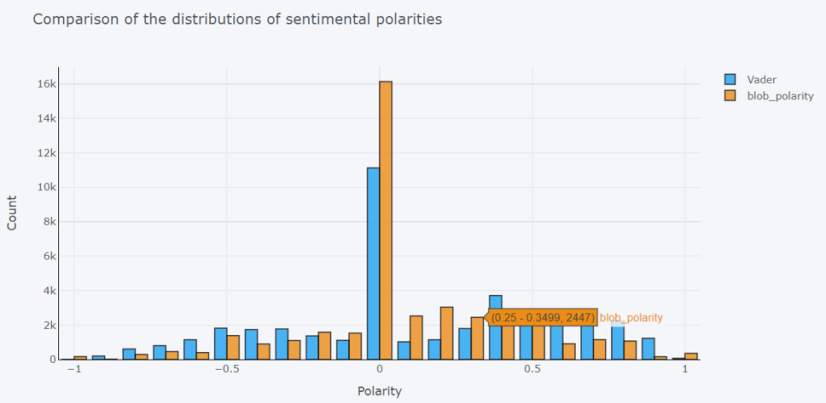


Fig 5: Distribution of Sentimental Polarities for VADER and TextBlob

Once we applied VADER and TextBlob libraries to the training corpus and extracted sentimental polarities for all documents, we observed that TextBlob marked 16,000 documents as neutral, while VADER marked 11,500 as

neutral. The accompanying figure illustrates that VADER has a wider range of accurately classifying documents as positive or negative compared to TextBlob.

Hypothesis: VADER-annotated documents will have better performance than TextBlob-annotated documents.

Lexicon Libraries	Precision	Recall	F1-Score
VADER	64%	68%	65%
TextBlob	63%	63%	62%

Table 3: Performance for Lexicon-based Approach (Unsupervised Sentiment Analysis).

4. Discussion

The aim of our project was to use different Supervised Machine Learning Techniques and perform a comparative analysis to understand how each model performs on the Twitter Data. We also aimed at using Unsupervised Machine Learning techniques to understand their usefulness in detecting the Sentiments in the given tweets.

For our Supervised Approach , we were inspired by *Alec et al* for the use of bi-grams and tr-grams for our Sentimental Analysis . After comparing the performance of our models on using bi-grams and tri-grams , we came to a conclusion that mono-grams and bi-grams based vectorization yielded better results and also we managed to outperform the model by *Alec et al.*
Our Logistic Regression model performed the best which made us realize that it's not always necessary to build complex models to get a good performance. Logistic regression being the simplest of classifiers was still able to outperform all the other models that we used.

In our Unsupervised Approach , we used dimensionality reduction methods and did a comparative analysis on different clustering algorithms to understand the effectiveness of this approach on the twitter data. We also used Lexicon Libraries like VADER and TextBlob to achieve better performance from our Unsupervised models. We came to a conclusion that the dimensionality reduction approach might not be optimal for twitter textual data .

5. Conclusion

We successfully implemented Supervised and Unsupervised methods to predict the Sentiment of Tweets with 82% accuracy . In future, we would like to improve our model performance by focusing on dealing with negations and emoticons present in the textual data . We would also like to implement Deep learning techniques on textual data by making the use of neural networks to train our model and perform a comparative study for the same.

6. Acknowledgement

This project was completed for CS6120 under the instruction of Professor Uzair Ahmad.

References

[1] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision.Technical report, Stanford.
[2] Kharde, Vishal & Sonawane, Sheetal. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. International Journal of Computer Applications. 139. 5-15. 10.5120/ijca2016908625.
[3] Qi, Y., Shabrina, Z. Sentiment analysis using Twitter data: a comparative application of lexicon- and machine-learning-based approach. Soc. Netw. Anal. Min. 13, 31 (2023). https://doi.org/10.1007/s13278-023-01030-x
[4] Luo and Tao, "Sentiment Analysis Based on the Domain Dictionary: A Case of Analyzing Online Apparel Reviews" 2018.