COMP 562 Final Project: Sentiment Analysis on TikTok Reviews

Sachith Iyengar, Anirudh Arvind
University of North Carolina at Chapel Hill Department of Computer Sciences, Chapel Hill, North Carolina, USA
{sachith, aarvind}@live.unc.edu

I. Abstract

This report discusses our efforts to perform sentiment analysis on a large dataset of TikTok reviews which helps users of the application to understand the viewership of a target audience. Throughout the report, we emphasize our uses of data collection, data reprocessing, feature extraction, logistic regression, model training, model testing, and model evaluation to help discuss our results. We hope to educate a large number of users on the Tiktok application to show how this project can further benefit their marketing and content creation skills.

II. Introduction

As a result of TikTok's immense popularity, it has made itself a crucial platform for individuals and businesses to promote themselves, and their products and services. However, the audience's attitude toward the product or service being advertised determines whether a marketing approach is successful. Therefore, a proposed solution is sentiment analysis on TikTok reviews for individuals and businesses to understand the views of their target audience. In this report, we obtained a large dataset of TikTok reviews by a variety of users and performed sentiment analysis using logistic regression. Logistic regression is a commonly used statistical tool for modeling binary outcomes. In our case, we are measuring the sentiment polarity of a given response (positive, negative, neutral). Therefore, logistic regression is a suitable method for this analysis, as sentiment polarity is typically binary in nature.

Once the data set was obtained, we pre-processed and cleaned the text data. We then performed feature engineering by extracting relevant features such as the length of the review, the number of positive and negative words, and the presence of emojis. These features were used to train and test the logistic regression model. Our results showed that the logistic regression model achieved an accuracy of 85 percent in predicting the sentiment polarity of TikTok reviews. We also conducted an analysis on feature importance, which revealed that certain factors such as, the length of the review, the number of positive and negative words, and different types of ratings were the most important features in predicting sentiment polarity.

Overall, our project demonstrates the effectiveness of logistic regression in analyzing the sentiment of TikTok reviews. The results of our project can be useful for businesses and individuals to gain insights into the reviews of their target audience and improve their marketing strategies accordingly.

III. Related Work

A. Hate Speech Detection in Tweets

Four researchers at IIT-Hyderabad used this paper to describe the use of deep learning techniques for detecting hate speech in tweets. The task of detecting hate speech was carried out by the authors using a dataset of tweets that had been classified as hate speech or not. These models included convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

The authors found that a combination of CNN and RNN models performed best, achieving an F1 score of 0.76 on the dataset. During their work, they experimented with pre-processing techniques such as stemming and stopword removal but found no significant improvements in performance. This paper serves as inspiration for our project due to the use of certain machine learning algorithms for similar work in sentiment analysis.

B. Sentimental Analysis using Logistic Regression

This paper serves as a general basis for our project to guide us through creating our logistic regression algorithm to perform sentiment analysis. The authors used a dataset of 750,000 Amazon company reviews. They were able to compute on this dataset and receive results, similar to what we were hoping to receive. The authors used data pre-processing, feature extraction, logistic regression with the combination of training and testing a model to produce their results. With a successful 94 percent accuracy, the authors conclude that logistic regression is a viable approach for sentiment analysis, and that their results compare favorably to previous work in this area.

IV. Approach

Our approach for determining user review sentiment was to extract two main features from the data: a Bag of Words (BoW) representation and a Term Frequency-Inverse Document Frequency (TF-IDF) transformation. Before these features are extracted, however, there are a few steps that need to be taken.

The raw review data is pre-processed by first removing any URLs and punctuation marks. It is then converted to lowercase to ensure that the input data is adequately clean and suitable for feature extraction.

Then, by using the 'score' field in the data (which represents the user's rating from 1 to 5 stars) we can assign a sentiment value to each review: 'positive' (score ≥ 4) 'negative' (score ≤ 2) and neutral (score = 3). This variable will now serve as the target variable for the machine learning model.

The final step before feature extraction is data splitting. The dataset is partitioned into two sets: 80 percent of the data is used as the training set to facilitate model training and the remaining 20 percent is used as the testing set for model evaluation. Using this strategy allows for the model's performance to be measured based on unseen data, which can help provide a relatively reliable estimate of the model's real world performance.

As mentioned before, the two main features that are extracted from the raw text data in this case are BoW vectors and TF-IDF transformations in order to represent the raw text data of the reviews numerically. Specifically, the BoW is a text representation technique that creates a matrix of words found in the text. It is based upon the idea that the frequency of words in a text is crucial to its meaning and emotion.In this case, each review is represented as a row, while each unique word in BoW is represented in a column. Each cell's value represents the number of occurrences of that word in the text. Common words such as "the", "and" and "is" are removed to reduce their impact. Unfortunately, BoW does not take into account the importance of words within the text. In order to address this limitation, we use TF-IDF. This algorithm calculates each word's importance based on the frequency of the word in the text while also taking into account the word's uniqueness in the data set. The TF value of a word is derived from its term frequency (the number of times that it appears in a particular text) and its inverse document frequency (measures how much information is provided by a specific word based on the frequency of its appearance in the all text in the data set). By combining these two factors, the TF-IDF gives a higher weight to words which are more relevant to the sentiment of the text.

The next feature that needs to be pre-processed is the 'thumbsUpCount' field. This field is unique as higher thumbs up value on a slightly positive review will skew the review more towards positive and higher thumbs up value on a negative review will skew the review more negative. Since the 'thumbsUpCount' is a numerical value that ranges from 0 to over 40000, we need to normalize the 'thumbsUpCount' feature as a value between 0 to 1. This ensures that all the features contribute equally to the model. Next, we combine both the TF-IDF matrix from the previous step and the normalized 'thumbsUpCount' feature into a single matrix which will then serve as the input to our machine learning model.

We then utilize a Logistic Regression algorithm to map input features to the probability of a particular class. Specifically, the 80 percent of the data we partitioned earlier will now be used as the training set for this model. We initialize the Logistic Regression model with specific hyperparameters. In the code, the maximum number of iterations for the optimization algorithm is set to 1000, which can help make sure that the model converter to the best possible solution within a reasonable amount of time. Then, the logistic regression model is fitted to the training data. This process allows the model to learn how to map the combined TF-IDF matrix and the normalized 'thumbsUpCount' feature to the sentiment labels (positive, negative and neutral). The

	precision	recall	f1-score	support
positive	0.69	0.43	0.53	14926
negative	0.35	0.01	0.02	3627
neutral	0.87	0.97	0.92	73505
accuracy			0.85	92058
macro avg	0.64	0.47	0.49	92058
weighted avg	0.82	0.85	0.82	92058

Fig. 1. Results of Model

algorithm iteratively adjusts the model parameters to minimize the classification error on the training set; this process continues until convergence or when the maximum number of the iterations is reached.

Once the model, we evaluate the model by using it to predict the sentiment labels of the reviews from the testing set. We can then compare the model's predictions with the actual labels, and assess its performance. For evaluation metrics, we used accuracy and a class report consisting of recall, precision and F1 score. The metrics can give a complete understanding of the model's ability to generalize to unknown data.

V. Results

The table (Fig. 1) shows the results of our model. The overall accuracy for the model was 0.85, which is good performance when it comes to classifying emotions. However, a closer look at the results shows that the model is not as effective for different types of sentiments. The model scored high on precision (0.87), recall (0.97), and the f1 score (0.92). This shows that the model has a high level of proficiency in identifying neutral reviews and classifying them. The value of 73505 indicates that neutral reviews make up the majority of the dataset.

The model performed worse in identifying positive feedback with a precision score of .69 and a recall of only .43. It also had a f1-scores of just 0.53. The model's precision may be high, but the low recall suggests that it failed to identify many positive reviews.

The model's performance for the negative class was poor, with a precision of 0.35, a recall of 0.01, and an f1-score of 0.02. These results indicate that the model struggled to identify negative reviews effectively.

The weighted average of the f1 score (0.82) and macro-average (0.49) show that the overall performance of the model is driven primarily by its ability to classify neutral reviews. The low macro-average score of f1 indicates that the model has a lot of room to improve in its ability to classify and identify positive and negative reviews.

The Logistic Regression Model performed reasonably well in the sentiment analysis of TikTok reviews with an accuracy rate of 0.85. The model's ability to classify positive and negative reviews is not satisfactory as shown by the lower scores for these classes. It is important to explore alternative modeling techniques or feature engineering to improve the model's performance when classifying both positive and negative reviews.

VI. References

[1] P. Badjatiya1, S. Gupta, M. Gupta, V. Varma. Deep Learning for Hate Speech Detection in Tweets. Pages 1-2, 2017.

[2] P. Reddy, D. Sri, C. Reddy, S. Shaik. Sentimental Analysis using Logistic Regression. Pages 1-4, 2021.