Sentiment Analysis of IMDb Movie Reviews

Kanakamedala Ragaja
Department of Data Science
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat 382007, India
202018012@daiict.ac.in

Shubhaditya Roy
Department of Data Science
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat 382007, India
202018021@daiict.ac.in

Aneri Thakkar
Department of Data Science
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat 382007, India
202018022@daiict.ac.in

Nisarg Thakkar
Department of Data Science
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat 382007, India
202018027@daiict.ac.in

Rushvi Shah
Department of Data Science
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat 382007, India
202018039@daiict.ac.in

***Abstract*—The aim of this project is to experiment with different machine learning approaches to classify dataset of IMDb movie reviews to perform sentiment analysis. Three different classifiers are used in this study which are: Logistic Regression, SVM and Naïve Bayes. The purpose of using different classifiers is to check for the best performances. The dataset includes movie reviews of 50,000 movies from IMDb. To improve the performance of our system additional features selection techniques, such as stemming, lemmatization, n-gram model, tokenization, and removal of stop words are applied. The performance of the system was evaluated using different metrics such as Final Accuracy, F1 Score, Recall Score and Precision Score. The proposed system achieved a maximum of 90% accuracy.**

Keywords—Sentiment Analysis, IMDb dataset, Logistic Regression (LR), Support Vector Machine (SVM), Naïve-Bayes, Term Frequency – Inverse Document Frequency (TF-IDF), N-grams Model, Recall Score, Precision Score.

I. Introduction

Sentiment analysis dig for people's opinions, views, evaluations, sentiments and emotions towards a product or service [1]. It analyses them and determines the attitude of people into the narrow domains such as: Negative or Positive [2]. There are numerous applications of sentiment analysis such as amazon recommends clothes based on customer reviews, on Snapchat and Facebook to recommend friends, advertisements [1], house rate prediction, movie ratings by IMDb or rotten tomatoes [2]. Sentiment analysis approaches to find negative, positive or neutral tone from the set of reviews and give phrases one of these three meaning. The crucial task of sentiment analysis is to take in consideration all the reviews of customer and calculate cumulative polarity of those reviews [2]. Reviews can be divided into 4 levels. First is document level, further divided into sentence level, which can be further categorized into word level, and then entity level [2].

The main purpose of our project is to analyse all reviews of a particular movie and try to predict the ratings of movie using sentiment analysis. Dataset used for this analysis includes 50,000 movie reviews from IMDb. Also, we have taken in consideration 3 types of classifiers logistic regression, SVM, Naïve Bayes which have different level of accuracy. Aim of this study is to learn which classifier promises most accurate results. Thus, our project includes comparative study between these 3 classifiers based on 4 metrics such as Final Accuracy, Recall Score, F1 Score, Precision Score.

II. Methodology

The various sentiment analysis approaches can be classified as: (i) Machine Learning (ii) Lexicon based and (iii) Hybrid approach[6]. The machine learning approach is used for predicting the polarity of sentiments based on training as well as test datasets. It applies machine learning algorithms and uses linguistic features.

The basic methodology of this project follows the machine learning approach for this Natural Language Processing (NLP) technique to determine polarity of the text using classifiers and a dataset that applies the classifier into sets of two namely, positive and negative. An organised method flow of the implementation is depicted in Fig. 1. Scikit-Learn, which is a well-known machine learning library tightly integrated with Python language and providing easy-to-interact interface, has been used for the implementation. The process commences with reading the text from the dataset, cleaning and pre-processing the data to prepare for the use of the machine learning algorithms. Since the text data cannot be given directly to these algorithms, it has to be converted into a suitable structured format. To improve the performance, various feature extraction and representation techniques have been used like tokenization, removal of stop words, stemming, lemmatization, term document frequency and N-gram vectorization. A system based on three different algorithms including Logistic Regression, Naïve-Bayes and Support Vector Machine (SVM) has been developed. Accuracy of the classifiers has been evaluated using different evaluation measures including F-Score, Precision Score, Recall Score and Accuracy Score.

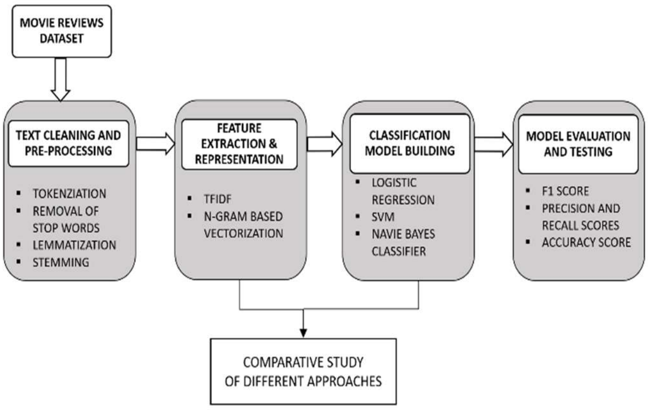
MOVIE REVIEWS DATASET
TEXT CLEANING AND PRE-PROCESSING
TOKENZIATION
REMOVAL OF STOP WORDS
LEMMATIZATION
STEMMING
FEATURE EXTRACTION & REPRESENTATION
TFIDF
N-GRAM BASED VECTORIZATION
CLASSIFICATION MODEL BUILDING
LOGISTIC REGRESSION
SVM
NAVIE BAYES CLASSIFIER
MODEL EVALUATION AND TESTING
F1 SCORE
PRECISION AND RECALL SCORES
ACCURACY SCORE
COMPARATIVE STUDY OF DIFFERENT APPROACHES

Fig. 1. Flow Diagram of Implementation

III. Classification Processing

A. *Data Collection*

The dataset used in this project is issued by the Stanford Artificial Intelligence Laboratory. They have collected the raw data from the IMDb website. They have provided a set of 50,000 reviews in total, the 50,000 reviews consist of 25,000 movie reviews for training and 25,000 for testing[5]. The dataset provided on their website is divided into two folders train and test, which serve as training data and testing data respectively. In each folder, there are two sub-folders pos and neg, to divide the data through different labels. In each of these sub-folders, there are multiple text files (12,500 to be precise) containing the content of movie review, with each file containing one review. The reviews are aggregated into two text files, one for the purpose of training and the other for the purpose of testing. The aggregation is done using the python glob library. The data is then loaded in a data frame to work on and is explored extensively.

B. *Pre-processing and Cleaning*

The process of converting a dataset into something that a computer can understand and work with is called as the pre-processing of the data. In a Machine Learning setup, if it is a natural language processing problem the text in the datasets invariably contain many useless symbols from the perspective of the computer, these symbols are removed using regular expressions which helps the programmer create a search pattern that helps match, locate and manage text.

During the pre-processing stage of this project the following things were mainly done: The data was converted into a structured format by creating a data frame with two columns, movie review and target. The target column was also defined as this project has used supervised machine learning algorithms. This column provides the label for the corresponding movie review column. The labels were defined as 1 and 0. If the movie review was positive, it was given the label of 0 and if the movie review was negative, it was given the label of 1. The labels were definitively provided as 0 or 1 based on the number of reviews, if the number of reviews in the aggregated dataset was less than 12,500 it was labelled as 1 as the negative reviews were aggregated first, the rest of the movie reviews were labelled as 0 which were positive.

Regex (regular expressions) library was used to remove punctuation marks, brackets, url's, html patterns and various other anonymous symbols from the movie review texts. A list of stop words aka a stop list was self-defined and the stop words were then removed from the movie review column of the data frame.

1) *Tokenization*

Tokenization is a fundamental procedure in natural language processing. Tokenization separates the text into smaller pieces of text called tokens. These tokens can be anything ranging from a character to a word or even a sub-word. Tokens are also known as building blocks for natural language. Tokenization is essentially done to create a vocabulary which refers to a set of unique tokens in the corpus.

In this project tokenization is done by splitting the text in the corpus and is used on n-gram vectorization models and tf-idf encoded models separately and the results are shown for each particular case i.e for each machine learning model in an exclusive manner.

2) *Removal of Stop Words*

Sometimes some very common words appear in the relevant dataset. These words which do not provide any value to the solution of the problem are called stop words. These words are usually aggregated in a list viz a stop list, the members of which are then discarded and removed from the vocabulary entirely during the pre-processing step.

In the project, a list of stop words aka a stop list was self-defined and the stop words were then removed from the movie review column of the data frame.

C. *Lemmatization*

Lemmatization is a process of normalizing the inflected forms of words. Homographic words cause ambiguity that disturbs searching accuracy and this ambiguity may also occur due to inflectional word forms. The process of lemmatization and stemming is similar with minor changes, while the benefits of both approaches are the same.

For example, words like “Talking”, “Talks” and “Talked” are the inflected forms of the word “Talk” which truncate the insignificant portion of words.

D. *Stemming*

Stemming is a process of removing commoner morphological endings from words in English. It is used to stem the words to its root words. For example, abate, abates, abated are stemmed to root word, 'abate'.

For example, “love”, “loving”, “lovingly”, “loved”, and “lovely” could all be used to illustrate they love something, which would be stemmed to “love”. Word stemming[9] is a task that chops each word down to its basic linguistic word stem form.

E. *Feature Extraction*

In order to avoid the problem of overfitting in any Machine Learning model. It is necessary to apply Feature Extraction techniques. Feature Extraction aims to reduce the number of features in a dataset by creating new features from the existing ones. These new reduced set of features should then be able to summarize most of the information contained in the original set of features. In this way, a summarized version of the original features can be created from a combination of the original set.

Out of the various methods available, this project uses

(i) TFIDF – Term Frequency – Inverse Document Frequency and

(ii) N-grams Model

1) TF – IDF

Tokenization Term document frequency refers to the count of specific words in the document and uses the concept of finding the term document using vectorizer.

In this project, the Term Frequency – Inverse Document Frequency (TF-IDF) vectorizer has been utilized to transform the text data into a feature set. It is the weighting factor of the document. It represents how much a word or term is important for the document. Supposing, if the counted value of the word "W1" is more than the counted value of the word "W2" in the document, the word "W1" is more important. So, the weighting factor is made much bigger in case of "W1" word.

2) N – Grams Model

An n-gram basically refers to any combination of sequenced words in a text having n tokens or words. If the N = 1, then it means a single word in a text, similarly if N = 2 then it leads to the combination of two sequenced words, and so on.

In this project, three different kinds of N-grams have been used in the classification to compute accuracies and generate different results. Supposing, if N-grams with different values of N based on the sentence "I like to do research" will be given as: (i) for N=1, Unigram, I, like, to, do, research

(ii) for N=2, Bi-gram, I like, like to, to do, do research and

(iii) for N=3, Trigram, I like to, like to do, to do research.

F. Classifiers

For sentiment analysis of movie reviews, this project uses three different classifiers in order to check and evaluate its performance based on metrics. The three different classifiers used are: Logistic Regression, Naïve Bayes and SVM.

1) Logistic Regression (LR)

Logistic Regression is an algorithm which belongs to exponential family. Logistic regression models are easy to interpret. Also, the data used in this project was very sparse, still this method worked very well. Linear regression model was also very fast as compared to other models [3].

In this project, logistic regression has been performed using TFIDF for text vectorization and representation and also n-gram model. The results are observed to be satisfying and slightly better in n-gram model case. Term document frequency refers to the count of specific words in the document and uses the concept of finding the term document using vectorizer.

2) Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm which stands for Support vector machines. It is known for giving widest difference between positive and negative samples. SVM is also known for its robustness and versatility [3]. SVM has proven its performance by giving good results in analysing various datasets.

In this project, SVM has been performed using TFIDF for text vectorization and representation and also n-gram model. The results are observed to be satisfying and almost similar in both cases.

3) Naïve Bayes

Naïve Bayes is used where it is assumed that one feature is not related to other [3]. It is based on Bayes theorem concept [4]. It is known for having less time complexity and thus being able to perform very fast even with very large datasets. It was simple to perform yet effective [3]. A Multinomial model of Naïve Bayes classifiers has been used.

In this project, Naïve Bayes has been performed using TFIDF for text vectorization and representation and also n-gram model. The results are observed to be satisfying and slightly better in n-gram model case.

G. Evaluation Metrics

For evaluation of the various classifiers used in this project various evaluation metrics such as Final Accuracy, Recall Score, F1 Score and Precision Score have been used for the comparative study.

Final Accuracy is the rate of correct classifications, which can be calculated as, the number of correct decisions the classifier makes, divide by the total number of test examples.

The Recall Score is said to be the fraction of the relative occurrences that were fetched. It is calculated as the ratio of tp/(tp + fn), where tp is the total number of true positives and fn is the total number of false negatives.

The F1 Score is the weighted average of Precision and Recall. Similar to Recall, Precision is calculated by tp/(tp + fp), where fp is false positive.

Thus, we can say F1 Score takes both false positives and false negatives into consideration. This Score is more useful when there is an uneven class distribution, and a good F1 score shows that the classifier has a good accuracy and have low false positives and low false negatives. F1 Score is given by, F1 Score = 2*(Recall * Precision)/ *(Recall + Precision).

As mentioned earlier the Precision Score can be calculated as, tp/(tp + fp). It gives accuracy of the model and shows how good the model is by showing the actual positives out of the predicted positives. It is the ratio of the accurately predicted positive values to the total predicted positive values.

IV. Comparative Study

The three different approaches used for implementation are machine learning models based on supervised learning technique which in general give higher accuracy results relative to Rule determined approach or Lexicon based approach for classification problems. A comparative study of sentiment analysis by using these machine learning approaches helps to identify a subjective and much enhanced model for a classification problem[7]. The comparison of these approaches was carried out by considering the key evaluation metrics which include Accuracy score, F1 score, Precision and Recall scores[8] obtained when these models built were tested on a common evaluation set.

On comparing the accuracy scores displayed in the Table-1 with respect to the two different text vectorization techniques used it can be observed that all the classification models that used N-gram model vectorization performed relatively. It is found out that the accuracy of the classifier Multinomial Naive Bayes classifier improved significantly in case of N-gram model.

	Evaluation Metric	Machine Learning Classification Technique Used		
		Logistic Regression	Support Vector Machine (SVM)	Multinomial Naive Bayes (NB)
TFIDF	Accuracy Score	0.88208	0.88328	0.81676
	F1 Score	0.88201	0.88520	0.75808
	Precision score	0.88152	0.88350	0.80533
	Recall Score	0.88152	0.88181	0.85887
N-gram Model	Accuracy Score	0.90080	0.90112	0.87020
	F1 Score	0.90848	0.90864	0.83160
	Precision score	0.90155	0.90185	0.86498
	Recall Score	0.89473	0.89517	0.90117

Table 1. A comparison of results from various sentiment analysis techniques

V. Results and Conclusion

It is evident from Fig 2 that the usage of N-gram model gives better results than TFIDF approach but it is to be noted that the N-gram approach requires more training time relatively.

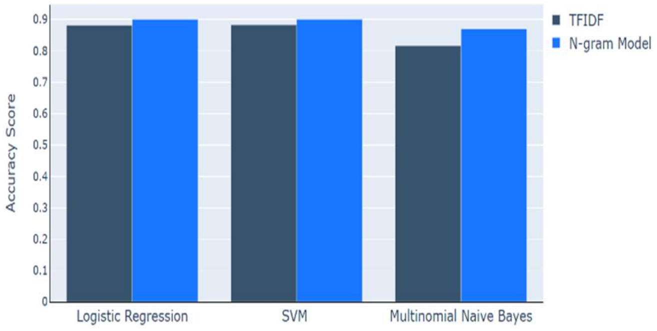

Fig. 2 Comparison of accuracy scores with respect to text vectorization techniques used

Among all the variety of approaches used n-gram approach with n-gram range of 1 to 3 in combination with SVM approach has given best results with an accuracy score of 0.9011 which in terms of percentage is 90.11% and performed better in comparison to the results obtained from other approaches.

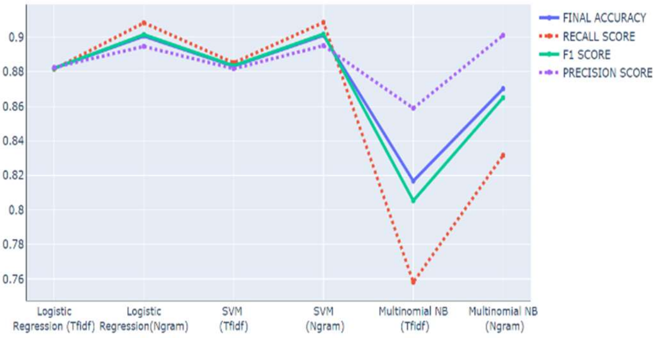

Fig. 3. Comparison of evaluation metrics obtained using different approaches

References

[1] Ang (Carl) Li ,"Sentiment Analysis for IMDb Movie Review"- *December 2019*

[2] Zeeshan Shaukat, Abdul Ahad Zulfqar, Chuangbai Xiao, Muhammad Azeem, Tariq Mahmood, “Sentiment analysis on IMDB using lexicon and neural networks” *Springer Nature Switzerland AG-2020*

[3] Swapnil Jain, Shrikant Malviya, Rohit Mishra, Uma Shanker Tiwary, “Sentiment Analysis: An Empirical Comparative Study of Various Machine Learning Approaches”, *Proc. of the 14th Intl. Conference on Natural Language Processing, 2016 NLP Association of India (NLPAI)*

[4] Hassan Raza, M. Faizan , Ahsan Hamza, Ahmed Mushtaq, Naeem Akhtar, “Scientific Text Sentiment Analysis using Machine Learning Techniques”, *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 12, 2019*

[5] Maas, Andrew L. and Daly, Raymond E. and Pham, Peter T. and Huang, Dan and Ng, Andrew Y. and Potts, Christopher,”Learning Word Vectors for Sentiment Analysis”, *Association for Computational Linguistics,Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*

[6] Alessia D’Andrea, Fernando Ferri, Patrizia Grifoni, Tiziana Guzz, "Approaches, Tools and Applications for Sentiment Analysis Implementation", *International Journal of Computer Applications (0975 – 8887) Volume 125 – No.3, September 2015*

[7] Devika M D, Sunitha C, Amal Ganesh, “Sentiment Analysis:A Comparative Study On Different Approaches”, Fourth International Conference on Recent Trends in Computer Science & Engineering. Chennai, Tamil Nadu, India

[8] Mohammad Mohaiminul Islam, Naznin Sultana “Comparative Study on Machine Learning Algorithms for Sentiment Classification” *International Journal of Computer Applications (0975 – 8887) Volume 182 – No. 21, October 2018*

[9] Tirath Prasad Sahu; Sanjeev Ahuja, “Sentiment analysis of movie reviews: A study on feature selection & classification algorithms”, *2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*