Sentimental Analysis and Topic Modeling on Movie Reviews

Youyou Xie
Georgetown Graduate School of Arts & Science
yx185@georgetown.edu

Abstract

In this project, various natural language processing methods have been employed to perform sentiment analysis and topic modeling. The entire dataset consists of 25000 training data with binary label and 25000 testing data without label. For the binary classification, I applied the bag of words followed by machine learning methods including Logistic Regression, K-Nearest Neighbors, Naïve Bayes, SVM, Decision Tree and Random Forest to compare the accuracy. The Logistic Regression model was used to label and test data due to its highest accuracy of prediction. Consequently, topic modeling was used to tackle the problem of topic classification, followed by categorizing the movie reviews into six categories and analyzing the distribution of them.

1 Introduction

Movie reviews serve as an important role to evaluate the quality of a movie. Although the numerical/stars rating to a movie on IMDB gauges the performance of the movie in some degree, a collection of movie reviews tell us about the deep qualities insight on different aspects of the movie. A kind of textual movie review containing an abundant of emotions and expressions are more powerful than a single score. Therefore, movie reviews give us more information to deeper analysis that can tell us whether the movie is catered to people's expectations.

Sentiment Analysis is a popular subject in natural language processing. Given a set of text, it aims to identify and extracts subjective information in those materials, helping us to better understand the textual meaning. A simple sentiment analysis algorithm can be used to determine the attitude of message and tell whether the underlying sentiment is positive or negative. There are many types of approaches for sentiment classification of texts, like using a machine learning method to do the classification.

Topic modeling is a form of statistical modeling, employing unsupervised and supervised machine learning techniques to discover hidden topics patterns in a large amount of unstructured texts. Latent Dirichlet Allocation is a widely used topic modeling techniques to discover hidden topic modeling techniques that used to classify text into different topics. By using this method, a mixture of the topics can be generated and prevalent topics distribution can be shown.

In this project, the sentiment analysis aimed to be used on a set of movie reviews given by reviewers and topic modeling was used to find out what topics are prevalent. In the first part, I tried to use very basic sentiment analysis technique. Like in many natural language processing tasks, the first step was to clean the texts and convert a sequence of texts into numbers, then I applied machine learning methods including Logistic Regression, K-Nearest Neighbors, Naïve Bayes, SVM, Decision Tree and Random Forest to compare their performance. After the validation test, the best classification method was selected to predict the sentiment label on test data. For the second case, the test data were used to do topic modeling by employing the LDA method, which improves classification by grouping similar words together in a small set of topics rather than using each word as a feature. In the end, the whole textual reviews were classified into six categories and their prevalent distribution was shown in the bar plot.

2 Methodology

2.1 Datasets

Movie reviews were extracted from the IMDB database, which is publicly available on Kaggle. [1]

[1] https://www.kaggle.com/c/word2vec-nlp-tutorial/data

The dataset contains 50,000 movie reviews and half of them are labeled with sentiment class: positive and negative. The sentiment of reviews is binary, meaning the IMDB rating less than 5 results a sentiment score of 0, and the rating greater than 6 have a sentiment score of 1. Neutral reviews are not included.

Initially, the dataset was divided into two subsets containing 25,000 examples for training and testing. The labeledTrainData.tsv has 25000 rows containing ids, movie reviews and their associated sentiment labels. The test.tsv has 25000 rows just containing ids and texts for each review.

2.2 Data Preprocessing

As a fundamental step for exploratory analysis, performing the data preprocessing is necessary to clean the texts. One necessary step prior to feature extraction was to removal of HTML tags like "
"x. The regular expressions matching were used to remove these HTML tags from the texts. After that, I also removed some non-letters and converted them to lower cases. Another important step was to make the textual information more meaningful that needs to remove unimportant words, like stopwords.

2.3 Feature Extraction

In order to convert a cleaned sequence of words to numerical features vectors. Two methods are applied for the extraction of meaning features from movie reviews for the following training purposes.

Bag of words: a simple and easy way to numerically represent texts. This method gathers the vocabulary from all the documents, then models each document by counting the number of times each word appears. In the IMDB data, the training dataset contains a large amount of vocabulary. Therefore, I selected 5000 most frequent words as the maximum vocabulary size to limit the size of the feature vectors.

TF-IDF method: the short for frequency-inverse document frequency, which is also a numeric measure that used to score the importance of a word in a document based on how often did it appear in that document. The institution for this measure is that: if a word appears frequently in a document, then it considered to be important and deserved the high score. But if a word appear in many other documents, it can not be considered as a unique word, therefore should be given a lower score. The measure formula is:

$$TF\text{-}IDF(t, d, D) = TF(t, d) * IDF(t, D)$$

Where t denotes the terms; d denotes each document; D denotes the collection of documents.

2.4 Comparison of Techniques

Having the feature vectors for each movie review, I applied several classification algorithms to evaluate and discovered the best algorithm with given training data. When evaluating the accuracy of algorithms, the training set cannot be used as a model because it may overfit and cannot predict useful information for the test data. Therefore, the original training dataset was divided to train set and test set to deal with this problem. 80% of given training dataset was used for training whereas the remaining amount was used for testing. However, there still exists overfitting on the test set, as the parameters can be compressed until the algorithm performs optimally. To tackle this problem, the rest of the training data should be used as a validation set. If the experiment seems to be successful, the final evaluation can be performed on the test set.

Since the results may be depended on a particular random choice of train and validation sets, k-fold cross-validation method was needed to improve this situation. This method is a resampling procedure used to evaluate how the model is expected to perform in general on limited data. In this workflow, it shuffled the dataset randomly, the split the dataset into 10 groups. The complete dataset was divided into multiple folds with different samples for training and validation each time and the final performance statistic of the classifier was averaged overall results.

Given below are the description of each evaluated method.

Logistic Regression: a linear model for classification rather than regression. It is the appropriate regression analysis to conduct when the dependent variable is dichotomous. It is a predictive analysis that used to describe data and to explain the relationship between one dependent binary variable and one or more independent variable.

K-Nearest Neighbors: a non-parametric method used for classification and regression. This method identify k nearest neighbors by computing distance

to other training records and used class labels nearest neighbors to determine the class label of unknown record. This method does not build models explicitly and unlike eager learners such as decision tree induction and rule-based systems.

Naïve Bayes: a set of supervised learning algorithms based on applying Bayes' theorem with naïve assumption of independence between every air of features.

SVM: a supervised machine learning with associated learning algorithms that analyze data used for classification and regression analysis. It uses known and labeled data to "train". It uses different "kernels" options such as linear, gamma, sigmoid and Gaussian. SVM can perform feature transformation into higher (or infinite Hilbert Space) dimensional space so that input vectors are separable by hyperplane. Most "important" training points are support vectors.

Decision Tree: a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features and to divide the data set into smaller data sets based on the descriptive features until you reach a small enough set that contains data points that fall under one label. It can handle both categorical and numeric data.

Random Forest: an ensemble learning method for classification. We make a single decision tree by randomly selecting subsets of the training data. Then, we repeat this process for many times to make a forest of a decision trees. To classify a new record, each tree offers a classification and the result is based on the majority classification based on all the trees.

After that, I make performance evaluation on validation dataset in each model to see which model performs best. I computed the standard performance metrics of Accuracy, Precision, Recall and F-measure.

	PREDICTED CLASS		
ACTUAL CLASS		Class = Yes	Class = No
	Class = Yes	True Positive(TP)	False Negative(FN)
	Class = No	False Positive(FP)	True Negative(TN)

Table 1

Accuracy(A) = $\frac{TP+TN}{TP+TN+FP+FN}$

Precision(p) = $\frac{TP}{TP+FP}$(Measure of sensitivity)

Recall(r) = $\frac{TP}{TP+FN}$(Measure of Specificity)

F-measure(F) = $\frac{2TP}{2TP+TN+FP}$

Since the Logistic regression algorithm performed the best grade, the sentiment label on testData.tsv was predicted by using the logistic regression method

2.5 Topic Modeling

In order to deal with high dimensions and large-scale multi-class textual data, topic modeling was applied to automatically organized understanding and summarizing categorical movie data.

LDA is an iterative algorithm. In the initialization stage, each word is assigned to a random topic. Iteratively, the algorithm goes through each word and reassigns the word to a topic taking into consideration. The intuition is that documents cover only a small set of topics and that topics use only a small set of words frequently.

After compared the bag of words and TF-IDF method, TF-IDF did not perform a good result, Then I used the bag of words to divide the whole 25000 test reviews into 6 topics and calculate the frequency of each topic.

At the end of the project, word cloud was used to draw the positive and negative figure about the labeled train dataset to analyze the frequency words in positive and negative reviews.

3 Results and Discussions

3.1 Sentiment Analysis

model	cross_val_score.mean	cross_val_score.std
Logistic	0.831950	0.006897

Regression		
K-nearest Neighbor	0.642800	0.010524
Naïve Bayes	0.793150	0.008527
SVM	0.832000	0.010085
Decision Tree	0.692300	0.012554
Random Forest	0.753350	0.007036

Table 2. Cross_validation_score in each model

As discussed above, I tried six classification models on the textual information in training movie reviews. From the training table, we can see that although SVM and Logistic Regression have the similar high accuracy rate about 0.832. The standard deviation of Logistic Regression is much lower than SVM'S standard deviation. Therefore, Logistic Regression model seemed to have best performance compared to other models with high mean accuracy rate 0.831950 and the small standard deviation about 0.006897. At the same time, k-Nearest Neighbors classifier had the worst performance with low mean accuracy rate about 0.64 and high standard deviation about 0.01. The general order of performance for the model was Logistic Regression > SVM > Naïve Bayes > Random Forest > Decision Tree > K-Nearest Neighbors.

Then I analyzed the matrix of Logistic Regression and classification report. The results were shown in below table.

	0	1
0	2003	465
1	369	2163

Table 3. the confusion matrix of Logistic Regression

	precision	recall	F1score	Support
0	0.84	0.81	0.83	2468
1	0.82	0.85	0.84	2532
avg/total	0.83	0.83	0.83	5000

Table 4. the classification report of Logistic Regression

3.2 Topic Modeling

```
#############run LDA using TF-IDF####################

Topic: 0
Word: 0.002*"stori" + 0.002*"love" + 0.002*"think" + 0.002*"episod" + 0.002*"charact" +
0.002*"best" + 0.002*"scene" + 0.002*"see" + 0.002*"seri" + 0.002*"action"
Topic: 1
Word: 0.002*"stori" + 0.002*"charact" + 0.002*"love" + 0.002*"think" + 0.002*"actor" +
0.002*"scene" + 0.002*"see" + 0.002*"come" + 0.002*"act" + 0.002*"plot"
Topic: 2
Word: 0.002*"charact" + 0.002*"think" + 0.002*"scene" + 0.002*"love" + 0.002*"play" +
0.002*"see" + 0.002*"stori" + 0.002*"look" + 0.002*"go" + 0.002*"thing"
Topic: 3
Word: 0.002*"stori" + 0.002*"charact" + 0.002*"love" + 0.002*"think" + 0.002*"life" +
0.002*"scene" + 0.002*"work" + 0.002*"actor" + 0.002*"see" + 0.002*"act"
Topic: 4
Word: 0.002*"love" + 0.002*"stori" + 0.002*"charact" + 0.002*"play" + 0.002*"think" +
0.002*"look" + 0.002*"see" + 0.002*"scene" + 0.002*"book" + 0.002*"origin"
Topic: 5
Word: 0.002*"think" + 0.002*"watch" + 0.002*"act" + 0.002*"see" + 0.002*"look" +
0.002*"scene" + 0.002*"charact" + 0.002*"horror" + 0.002*"thing" + 0.002*"plot"
```

Figure 1. Results of LDA using TF-IDF

```
############run LDA using bag of words################

Topic: 0
Words: 0.011*"charact" + 0.009*"stori" + 0.006*"book" + 0.006*"think" + 0.006*"play" +
0.005*"actor" + 0.004*"see" + 0.004*"come" + 0.004*"love" + 0.004*"cast"
Topic: 1
Words: 0.011*"love" + 0.010*"charact" + 0.008*"stori" + 0.008*"think" + 0.006*"go" +
0.006*"life" + 0.005*"thing" + 0.005*"see" + 0.005*"show" + 0.005*"come"
Topic: 2
Words: 0.008*"look" + 0.007*"think" + 0.007*"scene" + 0.006*"act" + 0.005*"go" +
0.005*"plot" + 0.005*"thing" + 0.005*"stori" + 0.005*"see" + 0.004*"kill"
Topic: 3
Words: 0.008*"scene" + 0.005*"go" + 0.004*"come" + 0.004*"look" + 0.004*"action" +
0.004*"play" + 0.004*"charact" + 0.004*"get" + 0.003*"make" + 0.003*"work"
Topic: 4
Words: 0.010*"funni" + 0.009*"think" + 0.009*"charact" + 0.007*"laugh" + 0.006*"comedi"
+ 0.006*"see" + 0.005*"look" + 0.005*"actor" + 0.005*"thing" + 0.005*"make"
Topic: 5
Words: 0.007*"stori" + 0.006*"year" + 0.005*"love" + 0.005*"perform" + 0.005*"see" +
0.005*"play" + 0.005*"best" + 0.004*"music" + 0.004*"work" + 0.004*"charact"
```

Figure 2. Results of LDA using Bag of Words

From the two figures shown above, I used two feature extractions to convert sequences of words into numbers before topic modeling and bag of words performed much better. Then I divided the whole text into six topics. The first topic may categorize to story plot. The second topic relates to love and romantic story. The third one has 0.006 act and 0.004 kill, maybe belong to horrible action section. The fourth one maybe associated with life scene plot. The fifth topic have 0.010 funny and 0.007 laugh, so it categorizes to comedy movie. The last one related to work and year, it can belong to epics and historical plot.

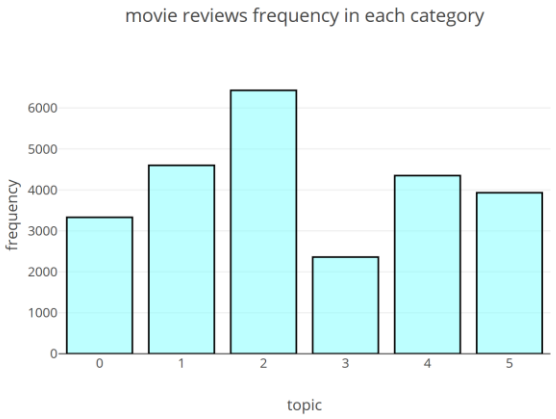

Figure 3. Movie reviews frequency in each topic

After analyzing the movie topics, I draw a bar plot to see what topics are prevalent in testing movie dataset directly. People prefer to see movies relating to horror, action or exciting films. Those movies about life scene plot may become less popular.

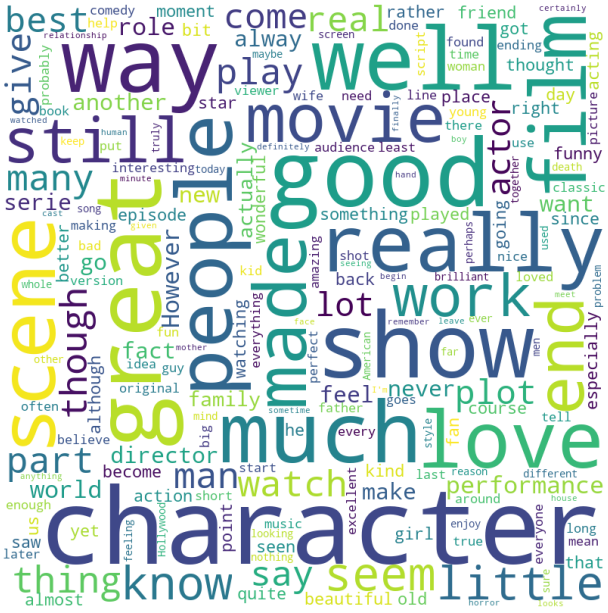

Figure 4. word cloud of positive words

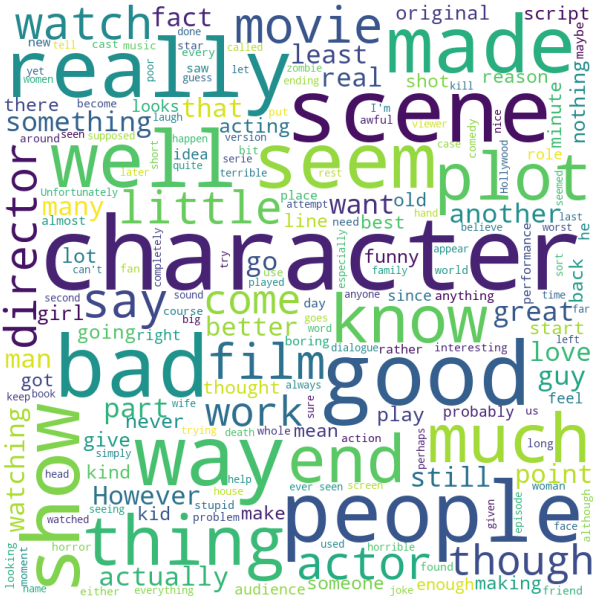

Figure 5. word cloud of negative words

The figure 4 is a word cloud about the positive sentiment in the training dataset and the figure 5 is about negative. There are many positive words like good, love, great, best shown in the positive figure. However, in the negative picture, although it really exists some positive words like good, well, it contains some discouraging words like bad, however, actually, little, least.

4 Conclusion

In this project, I learned a lot of knowledge about natural language processing and took them into practice. My exploratory consisted of two main parts.

In the first part, the movie review dataset about IMDB was used for analyzing and the bag of words method was applied to represent words numerically. Several kinds of classification, including Logistic Regression, K-Nearest Neighbors, Naïve Bayes, SVM, Decision Tree and Random Forest, were carried out to perform binary classification. Since Logistic Regression performed the best accuracy result, this method was held out to classify the rest of the unlabeled dataset.

In the second part, the topic modeling was implemented to classify the test movie reviews and divide them to six topic models. Those topics were applied to the whole test dataset to observe each movie review belonged to which topic. After that a bar was plotted to see which topic is the most

prevalent. In the end, the word clouds were drawn to see the common words that appear in positive movie reviews and negative one.

My exploratory is just the beginning. There are many areas to be discussed which needs more time and effort. Understanding of movie reviews is interesting because I can not only know people's attitude towards movie but also the popular movie section behind the data. Since the movie reviews classification is a great strategy to understand people's feeling. If I can find a way to see the similarities and differences among those reviews, people's favorite expectation on movies can be detected and predicted. I hope my study proposes an intriguing topic in this area and a way to study it.

References

Movie Reviews Sentiment Analysis with Scikit-Learn - http://www.pitt.edu/~naraehan/presentation/Movie+Reviews+sentiment+analysis+with+Scikit-Learn.html

Large Movie Review Dataset - https://www.kaggle.com/c/word2vec-nlp-tutorial/data

Shashank Gupta. 2018. Sentiment Analysis: Concept, Analysis and Applications

Topic Modeling - https://en.wikipedia.org/wiki/Topic_model

LDA - https://ai.stanford.edu/~ang/papers/nips01-lda.pdf

Bag of words - https://en.wikipedia.org/wiki/Bag-of-words_model

Kunlun Li, Jing Xie, Xue Sun, Yinghui Ma, Hui Bai. Multi-class text categorization based on LDA and SVM

Hadi Pouransari and Saman Ghili. Deep learning for sentiment analysis of movie review

Ankit Goyal and Amey Parulekar. Sentiment Analysis for Movie Reviews

Susan Li. Topic Modeling and Latent Dirichlet Allocation(LDA) in Python

Asiri Wijesinghe. October 2015. Sentiment Analysis on Movie Reviews

V.k. Singh, R Piryani, A.Uddin, P.Waila. Sentiment Analysis of Movie Reviews-A new Feature-based Heuristic for Aspect-level Sentiment Classification

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng and Christopher Potts. Learning Words Vectors for Sentiment Analysis

Aditya Timmaraju and Vikesh Khanna. Sentiment Analysis on Movie Reviews using Recursive and Recurrent Neural Network Architectures.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. manning, Andrew Y. Ng and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank