HW2

Vectorization

Bing-Je Wu

IST 736 – Text Mining

Table of Contents

[Introduction 2](#_Toc37344347)

[Analysis and Models 2](#_Toc37344348)

[About the Data 2](#_Toc37344349)

[Data Processing 2](#_Toc37344350)

[Results 2](#_Toc37344351)

[Conclusion 4](#_Toc37344352)

[Reference 4](#_Toc37344353)

# Introduction

Movies, also called motion picture, play a crucial part in daily life, especially in the 20 century. Everyone loves watching movies. Movies are full over the places. People can rent a movie from the supermarket such as Walmart, Shoprite…etc. And with the improvement of the technology, the way to get the access to a movie has become easier and simpler. People can watch a movie through internet. Many companies, such as Netflix, Hulu, Amazon, Disney and Apple, has provided the streaming services for people to watch movies at home without going out to pick up a DVD or the old fashion of VHS. With that being said, it comes to another issue. Since people get to watch a movie easily, the volume of comments, reviews or critics about a movie has also been increasing exponentially this day. Having the ability to decipher the thought of the movie review becomes one of the major tasks in text mining. This is why more and more applications are built to solve the question.

Analysis and Models  
os module and pandas library were used for loading text files. CountVectorizer from scikit-learn package was implemented to transform text into vectors.

## About the Data

Data (<https://github.com/bing020815/Syracuse-University/tree/master/IST-736/HW2/data>) were sampled from Large Movie Review Dataset (<https://ai.stanford.edu/~amaas/data/sentiment/>) as a preliminary work on course project. The movie reviews were originally collected from IMDB, an online database of information related to films, television programs. Any review with the stars rating greater than or equal to 7 is labeled as positive review. If the rating is less than or equal to 4, it is labeled as negative review. The reviews with stars rating between 5 stars and 6 stars were left out. The sampled dataset contains 40 movie reviews in total. 20 reviews were labeled as negative review and the other 20 reviews were labeled as positive review. Each review was stored in a text file. 10 negative reviews and 10 positive reviews were put under the ‘train’ folder. And the rest of the reviews were put under the ‘test’ folder.

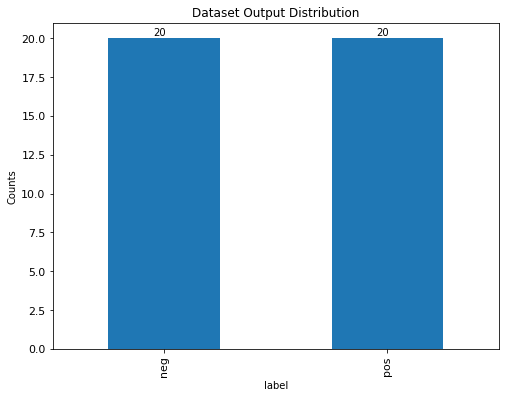
Data Processing  
The dataset was stored in text file under two folders, ‘train’ folder and ‘test’ folder. To read the data into Python, an os module was applied to get the complete path of files and file names. With the complete path and file names stores in list variables, A CountVectorizer has been instantiated with the parameter setting, input='filename', to convert the textual data into sparse matrix. Then, a method, toarray(), was used along with a pandas function, DataFrame, to create a pandas dataframe. The column names were obtained through the method, get\_feature\_names(), from the instantiated CountVectorizer object. An additional column, ‘label’ was added by applying a lambda function to get the sentiment based on the filename list.

# Results

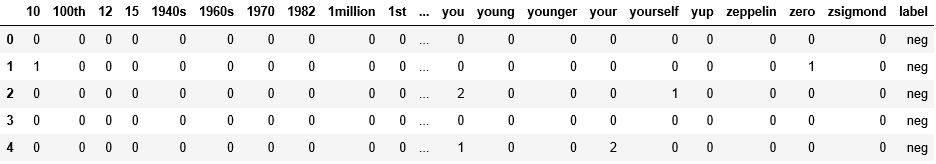
The dataset has been cleaned and converted into pandas dataframe that is ready to for many data science tasks, such as creating a plot, running statistics analysis, or building a model with packages. Figure1 shows the distribution of the output, 20 negative counts and 20 positive counts. It is consisting with the number of the data that sampled from the Large Movie Review Dataset. Figure 2 shows a snapshot of the dataframe. The last column, each row represents a document, movie review. And, the ‘label’ column shows the sentiment of the document/review. The rest of the columns are tokens that CountVectorizer learned from the dataset.

A default setting was mainly used for vectorization except “input='filename’”. Text were converted to lowercase and vectorized to tokens. Each of tokens represents a vocabulary or word that may or may not be used for each of documents/reviews. This dataframe is also called document term matrix. It is highly used in the text mining tasks. For example, the scikit-learn package support pandas dataframe. Many of algorithms from scikit-learn package primary use pandas dataframe as its input. Thus, having the textual data into a structure format, pandas dataframe in this case, is the crucial step when people start to work on text mining process. And, on top of that, pandas library provides tones of method that can be in aid of analysis. It can do pivot table on the structured data. It can be used to provide visualization through other packages such as, seaborn, matplotlib…etc.

**Figure 1:** Distribution plots for movie reviews.



**Figure 2:** Sample dataframe



# Conclusion

Deciphering text content could be hard. Text are everywhere around the world. And, needless to say, extracting the information from tons of movie reviews requires a lot of works. If the tasks are performed manually, it could cost badly in time and money. A smart and efficient way to extract extra information for a business is to apply text mining techniques. In that way, a company can have the direction of which movie genre is the most attracted to the public. Then, a low risk of investment can be made in the production.

To perform text mining tasks efficiently and correctly, many skills and techniques should be learned and mastered to achieve the objectives. Among all the skills and techniques, getting structured data takes majority of time and effort. With the structured data in hand, it can make many impossible to become possible. And that is why people spend time on developing tools and packages to help users get unstructured data into structure format for performing text mining tasks.

As a data scientist to perform a task on text mining, the technique of transforming the unstructured data into structure format is crucially important. Once the step has been done correctly, there are infinity of applications that can be developed or created for helping or contributing the society.

# Reference

1 Movie Review Dataset. [Web Link](https://ai.stanford.edu/~amaas/data/sentiment/)   
Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).;  
2 Dataset. [Web Link](https://github.com/bing020815/Syracuse-University/tree/master/IST-736/HW2/data) ;   
3 Film, from Wikipedia. [Web Link](https://en.wikipedia.org/wiki/Film) ;  
4 IMDB, from Wikipedia. [Web Link](https://en.wikipedia.org/wiki/IMDb)