**STATS 402 - Interdisciplinary Data Analysis**

**<Project Title>**

Milestone Report: Stage 2

**< lian zeng, yechen wang, zehcen yuan >**

**<**[**lz166@duke.edu**](mailto:lz166@duke.edu)**,** [**yechen.wang@dukekunshan.edu.cn**](mailto:yechen.wang@dukekunshan.edu.cn) **,** [**zy96@duke.edu**](mailto:zy96@duke.edu)**>**

**There are no specific requirements for the stage 2 report since the progress may vary among different groups. Generally, there are four parts you need to cover in your report.**

1. **The current status of your project**

**For example, the detailed techniques you adopted to conduct the project. Has your group made any technical route adjustments? This part is essential for the groups whose actual adopted method is different from their milestone report 1. You need to explain the reason for the change.**

**2. Demonstrate some initial data preprocessing results if you have.**

**3. The plan for the next two weeks.**

**Data**

**Demographic Filtering (DF)**

We have already finished the first part of our recommendation system: demographic filtering. Demographic filtering will provide recommendations based on the movies’ popularity and rating voted from users.

The basic mechanism for this filtering method:

* Calculate the count of votes and the total score of ratings each movie received.
* Calculate the average score of ratings.
* Combine the popularity and average score of ratings.
* Sort the scores.
* Recommend users movies with the high ratings.

In this way, we have made the first recommendation system based on demographic filtering. Finally, we output the original title, language, overview and score of the top 5 movies.

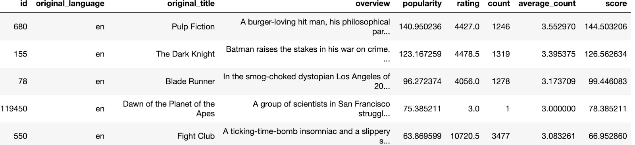


Figure 1 top 5 movies (Demographic Filtering)

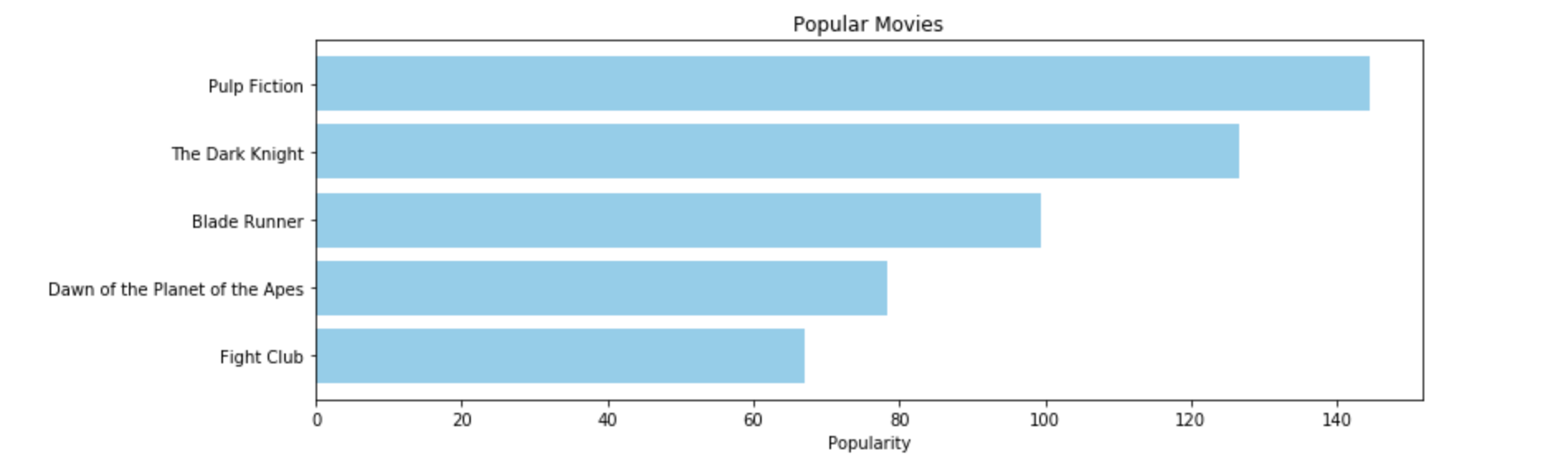


Figure 2 top 5 movies (Demographic Filtering)

**Initial data preprocessing results**

1. The first step of data preprocessing is cleaning the dataset. We chose to keep the features of the films that were useful to us.

Our movie dataset has the following features:

* id: unique identifiers for movies.
* original\_language: The language in which the movie was created.
* original\_title: The titles of the films before translation.
* overview: Descriptions of movies.
* popularity: A number that specifies the popularity of a movie.

The chart 1 below is part of our pre-processed dataset.



Figure 3 pre-processed dataset

2）The second step is to clean up the movies which miss some features and only keep the English movies for further purpose.

**Content Based Filtering (CB)**

The idea behind CB is to cluster the data into several groups and if a user likes an item in one of these groups, then it is likely that the items clustered into the same group will be attractive to the user as well. In our project, we want to do the clustering based on the overviews of each movie.

In traditional NLP models, it is common to use one-hot encoding, that is, each word in the vocabulary is represented by an N-dimensional one-hot vector, where N is the size of the vocabulary. For instance, if a vocabulary contains 5000 words, then N = 5000. All entries of the vector are 0 except the i-th element with value 1, where i is the index of the word in the vocabulary. Take a vocabulary contains 5000 words whose first word is “a” and last word is “zoo” as an example, “a” will be represented by a one-hot vector with zero entries except the first entry with 1 as its value and “zoo” will be represented by a one-hot vector with zero entries except the last entry with 1 as its value.

This one-hot encoding technique is simple but ignores the similarity between words. We employ Word2vec, a word embedding technique mapping the words into a vector space considering the context of the word.

In order to train a Word2vec model, we use the Continuous Bag-of-Words Model (CBOW) architecture proposed by Tomas Mikolov et al. [1], which uses the context to predict the current word.

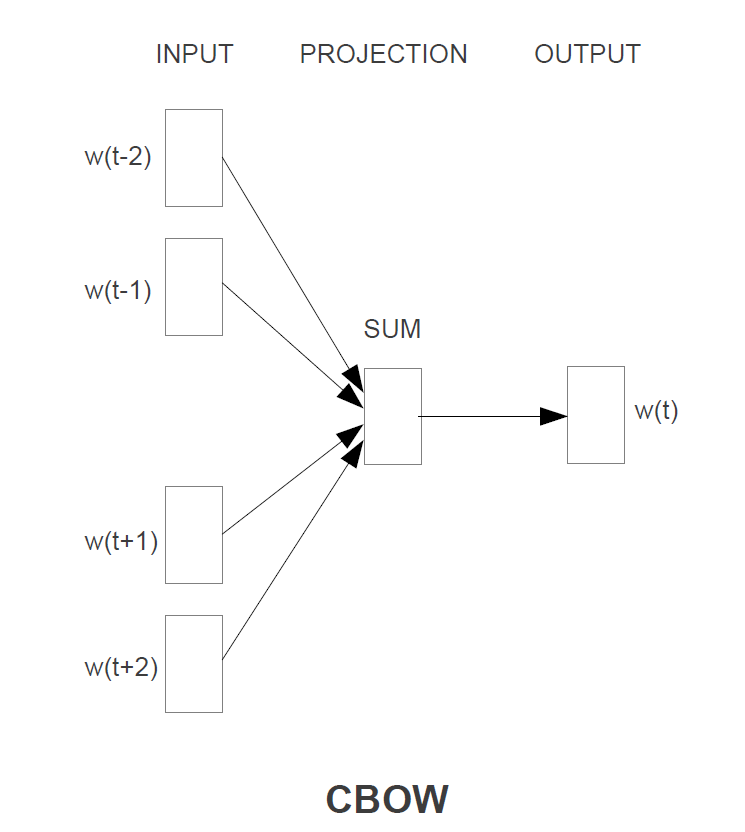


Figure 4 CBOW architecture

The input of the network is the context of the targeted word and the output is the targeted word. In our model, we use a 25-dimensional vector to represent each word in the vocabulary.

Besides the model we trained, we will also use other Word2vec models online that trained on larger vocabularies. We will later compare the results of different Word2vec models by checking the performance of the models on our movie dataset.

The naïve but commonly used methods to embed a paragraph or a sentence is to use the maximum, minimum or average of all words in the paragraph or sentence [2,3]. In order to further cluster the overviews, we use the average method. First, we take the sum of vectors of all words in an overview and then take the average of the sum. The outcome is treated as the vector representing of the current overview. Later, we will cluster the overviews by these vectors.

**Future Plans**

# References

1. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space
2. Tang, D.; Wei, F.; Yang, Y.; Zhou, M.; Liu, T. and Qin, B. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification, the 52th ACL conference, Baltimore, Maryland.
3. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.; Ng, A. and Potts, C., 2014. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, EMNLP 2014.