**STATS 402 - Interdisciplinary Data Analysis**

**< Recommender system for Social Media >**

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)**>**

In this milestone report we will introduce the current status of our project. This report is organized as follows: In section 1 and 2, we will introduce the progress we have made in constructing our system. Since our system consists of three parts and we are currently working on the first two parts, we will introduce the progress of part 1 in section 1 and the progress of part 2 in section 2. Section 3 will indicate the difficulty we are facing and what could be the current solution. And finally in section 4 we will explain our future plans.

1. **Demographic Filtering (DF)**

We have already finished the first part of our recommender system: demographic filtering (DF). DF 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 recommender system based on demographic filtering. Finally, we output the original title, language, overview and score of the top 5 movies.

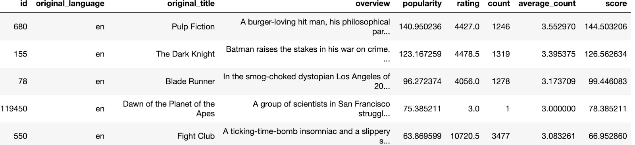


Figure 1 top 5 movies (Demographic Filtering)

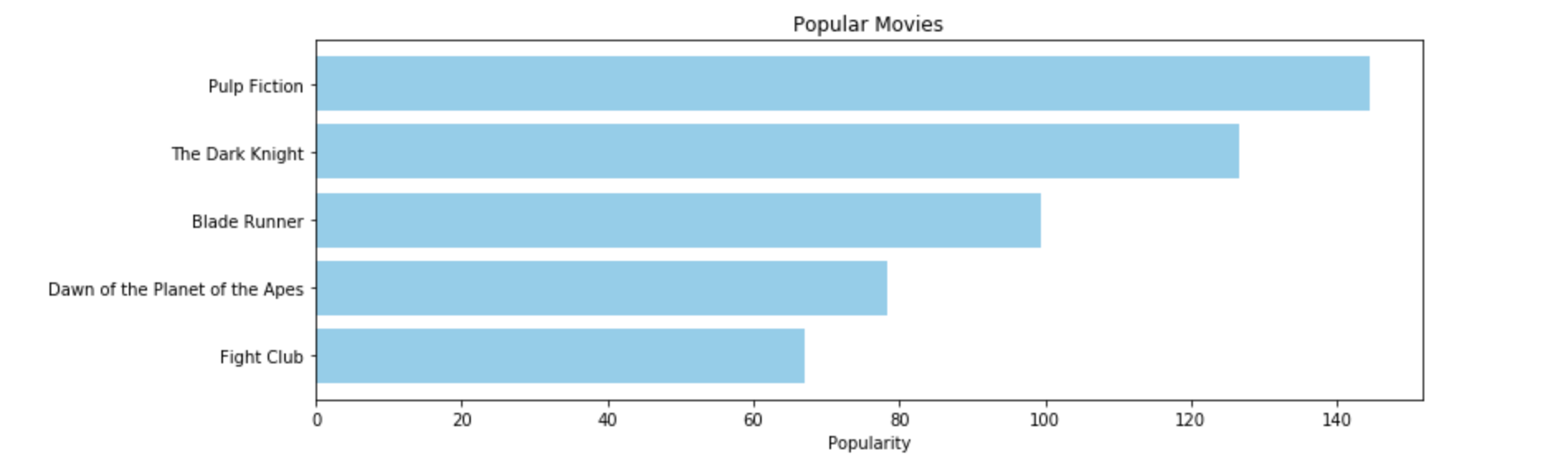


Figure 2 top 5 movies (Demographic Filtering)

**Initial data preprocessing results**

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 figure below shows part of our pre-processed dataset.



Figure 3 pre-processed dataset

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

1. **Content-Based Filtering (CB)**

The idea behind Content-Based Filtering (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 it ignores the similarity between words. Instead, 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.

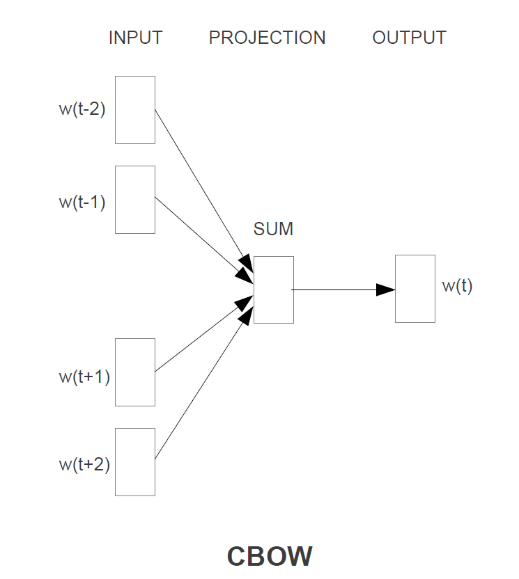


Figure 4 CBOW architecture

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.

1. **Difficulty and Current Solution**

The main difficulty we currently face is how to measure the performance of our approach. Because the quality of the results of a recommender system really depends on each individual, it is hard to find a general method to qualify how good the results are. We have also done some research online, but there seems to be no universal way to measure the accuracy of a recommender system. Also, we are not trying to make our recommender system 100% accurate. Because users still want something that is new to them. And if the accuracy rate is too high, it may raise some concerns about user privacy. So, the biggest difficulty for us is not being able to find a common and reasonable metric for our system. One current solution is to observe the results produced by our system and make an intuitive and subjective judgment. This approach is reasonable because the recommender system itself and the results are heavily influenced by subjective determinants, and we believe it is a viable approach for our situation.

1. **Future Plans**

Our future plan for the next two weeks is based on the current progress of our project. Our project is divided into three parts. For the recommender system based on DF, it is already finished. For the recommender system based on CB, our current solution based on word2vec is a little bit naive. We plan to try other methods such as TF-IDF, sentence2vec and attention model. Comparison will be made through all those different techniques and we will analysis their performance according to our experimental result to show their advantages and disadvantages under different circumstance. For the collaborative filtering (CF) system, we will start to build our model based on transformer. Few methods will be investigated, and we will choose the best model according to our own observation. In the next two weeks 3 parts of our system will still be optimized separately. And we will put all those things together to build a complete system in the final week.

# 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.