Movie Search Engine Based on Description

Dongjian Chen, Ren Wang, Yumeng Shao

December 11, 2019

Abstract

Keywords

Movie, BM25+, Information Retrieval, Search

1 Introuction

Watching movies has become one of the most popular ways of entertainment and there are hundreds and thousands of new movies produced every year. An average person could have watched dozens or hundreds of movies. It is not rare for someone to bump into an annoying situation when they could only remember a rough story or some segments of the movie, but just could not recall its name.

In this project, we aim at helping people to retrieve the film they want when they forget the title and could only remember a part of the story. We have designed and implemented a search engine to find corresponding movies with description or keywords provided by the user.

2 Design Description

2.1 Dataset

We obtained our data from two sources:

Kaggle movie dataset: (`https://www.kaggle.com/rounakbanik/the-movies-dataset#movies_metadata.csv`)

IMDB website (`https://www.imdb.com/?ref_=nv_home`)

2.2 Data processing

In this project, our retrieval system is built to retrieve movie by its description. Our data document for each movie contains its overview and keywords related to it.

We first scrape keyword data from a website and combined it with the overview of each movie from the Kaggle dataset into one string. Then, we tokenize the string into words and do stem and lowercase on each word. Finally we remove stopwords and punctuations, and the document is then ready for query. The above mentioned steps are also applied to query sentences.

2.3 Model

We tried several models and observed their performances to choose the best one. The score function in final model we used is BM25+ function (Inspiration from `https://www.eecis.udel.edu/~ypeilin/pub/ictir2016_long.pdf`) which is :

$$S = \sum_{t \in q} \frac{(k_3 + 1) \cdot c_t^q}{k_3 + c_t^q} \cdot \left[\frac{(k_1 + 1) \quad \cdot c_t^d}{c_t^d + k_1 \cdot \left(1 - b + b \cdot \frac{l_d}{L}\right)} + \delta \right] \cdot \ln \left(\frac{N + 1}{N_t} \right)$$

This BM25+ function add two more parameters to change the weight of TF and IDF in the function and it make our performance better in our test query set.

2.4 Interface

We have also built an interface for users to Input the description they remember.

3 Results and evaluation

We formulate 35 short descriptions of movies and use our search engine to fetch the result. Then we check the rank R of the expected movie. We can then define 1/R as the precision of our search engine.

We calculate the harmonic Mean of R (if the expected result isn't retrieved, R is infinity.) and use it as the evaluation for the search engine.

4 Conclusions

The result of this project is generally satisfying with our experiments with it. We succeeded in returning the related results with the queries and has improved its performance with data processing and parameter tuning. Thus, with our search engine, people can mostly get the movie they referred to. However, the search engine might not work well with subjective feelings on movies since the data we used for query are neutral descriptions.

5 Future Improvements

During data processing, we have tried part of speech tagging on the sentence and give more weights to certain types. However, it doesn't turn out well as we expected. We believe that digging further into this by analyzing the parse tree to evaluate the part of speeches more precisely can help with giving better weights. Moreover, we believe that keeping a search and view log can also help with improving accuracy. Within a series of similar searches, we can analyze the movies the user has clicked into and find the similarities to feed back into the algorithm and adjust the weights accordingly.

6 Acknowledgement

Prof. Qiaozhu Mei, Mr. Wei Ai from University of Michigan; Teaching Assistant [Shiyan Yan, Xinyan Zhao] for SI 650 at University of Michigan.

References