Script-based Film Recommender

Xinyu Tian (xt5), Muxia Yi (muxiayi2), Luning Wang (luningw2)
https://github.com/tianxiny/script-based-film-recommender

This project aims at developing a novel film recommendation system which is script-based recommender. Generally, the film recommendation system can be built based on popularity, content or collaboration. Most content-based applications and websites suggest users the movies or TV series they will like based on the genres and the keywords in comments to obtain the similarity. However, films by the same genre may have entirely different plots and in various styles. In this project, we attempted building a film recommender based on scripts so that the users can find the films to their tastes based on the scripts of their favorites. Also, as a search engine, they can retrieve the most relevant scripts by entering keywords. Our tool might be more useful to TV script writers who can be inspired by related works, movie reviewers who want to cite the lines spoken in several movies, or film buffs who enjoy the movies with specific elements. In the project, we used a corpus containing 1,068 films created by UC Santa Cruz.

Functions

This part provides the overview functions of our recommender, which has two main functions: film recommendation and search engine.

Film recommendation

We provide two entry modes. The user may enter either a block of script or a couple of words of their favorite film. The tool will return the top 10 most similar films, the similar genres predicted by classification models, and the common topics of those genres extracted from the database.

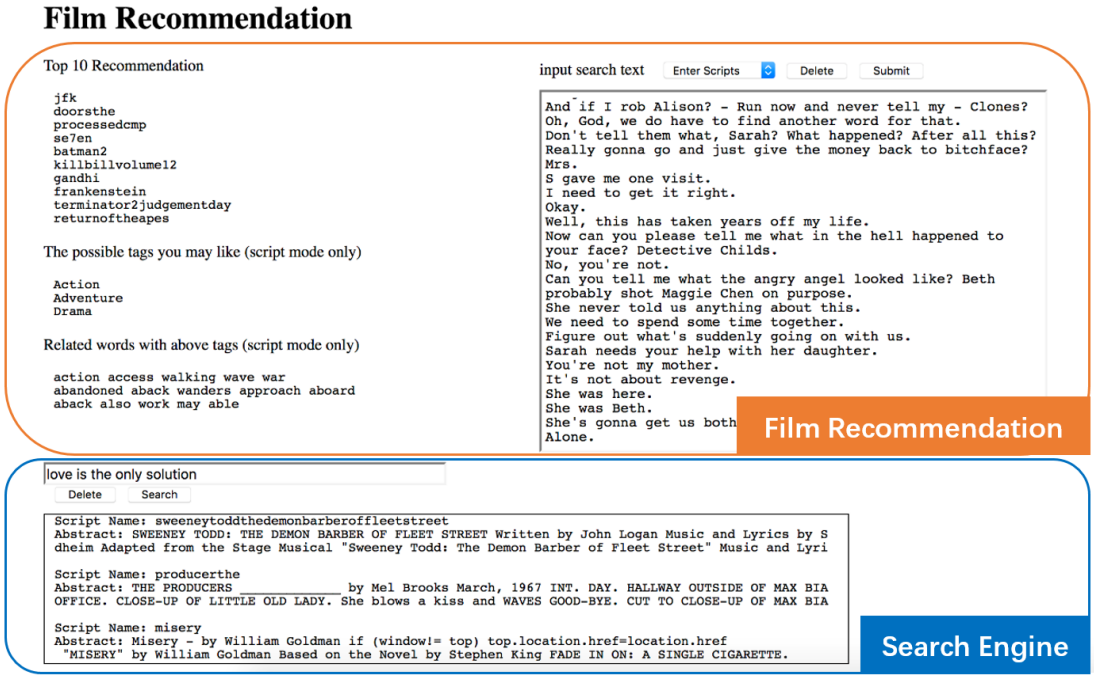
Our tool can recommend movies or related TV episodes based on the content of the scripts rather than only the tag given to the whole category, examples can be seen in the figure below. The recommendation would be a different episode in the same series or a related episode from a totally different series. This is a function which no known tools are doing at present. The new tool, which is based on contents of scripts, will not only have tags, but also have context comparing functions, so it can recommend really “right” movies to people.

Search Engine

Another main function of our recommender is that it also can be a search engine. Given a sentence or a question entered by the user, our tool can return the most relevant films h/she might like in the database and allow the user to view the head of their scripts.

This function can solve the problems that people always receive a disliked search results when film websites return movies by tags. For example, some TV series, just like Black Mirror, has no connection between each two episodes, so a person who likes episode 1 may not really like the topic in episode 2. However, our search engine

would return the similar topic since our tool is based on scripts. For example, if a users serch Kong King 1, then our tool would return Jurassic Park I.



Implementation

Database (MySQL)

Here we use MySQL as the database where it can store the user information. This feature is not fully implemented because in our system multi-user feature is not used. In our case we use MySQL to store the topic words of each category of movies.

Search engine (MeTA)

In this part we use MeTA to build the search engine. It takes the input sentence of the user and return the most relevant scripts according to the ranking. Because MeTA is able to send requests to several different search engineers at the same time, in this project, we use MeTA with PL2Ranker, which is significantly efficient. In addition, we also use different models in MeTA to do the search engineer part.

Quality words extraction (AutoPhrase)[1]

Through extracting high-quality phrases or texts from scripts of movies or TVs by implementing phases mining tools, we are able to process the data by extracting quality words of different granularity and phrases from it using software.

Since nearly every script contains many phases, the workload would be pretty different when applying different methods of phases mining tools. In this project, after carefully check several data-driven methods for extraction of phrases, AutoPhrase is implemented to extract the high-quality phrases or words with limited human works.

Tag classification (SVM)

We use supervised learning to tag the corpus (in script mode) and give three most related tags. In the dataset, all the scripts are already tagged. So we extract keywords from them using AutoPhrase and use the keywords as features, tags as labels. It is noticeable that some scripts belong to multiple tags. Then we use SVM to train our model and make the prediction for the input corpus.

Since in practice, some scripts belong to different tags in a binary way, so using Support Vector Machines (SVM) would be efficient for this task since SVMs are binary classification. In order to find a high efficient classification with less human effort, we established mathematical model, and used optimization algorithms including Stochastic gradient descent and coordinate descent methods for SVM classifiers when analysed scripts with using tags and comparing to other source and finally decided use traditional SVM for the input corpus, the parameter gamma of SVM is 20.

Topic Modeling (NMF)

The purpose of the topic modeling is to identify the common topics of the films by the same genre to provide users with the hints of plots they may enjoy or the shared flavors of the film by the genre. It is followed to the prediction of genre tags by the script. Prior to implement the topic models, the normal NLP procedures were performed by using NLTK, including tokenization, stopword removal and stemming (we skip stemming temporarily for better results). Then, the words are converted into vectors and normalized by TF-IDF. Here, we set the max_df at 0.85 and min_df at 0.15 to filter out the topic words at two tails. The model was trained with the help of scikit-learn.

Actually we compared the performance of different topic modelings, including LDA, NMF, DMM, BTM and etc. We finally selected NMF since the results are more meaningful to human judgments even though they are not very coherent. To avoid unnecessary delay in running the topic modeling, we extracted the topic words for each genre and stored them in a .txt file to read.

Web design (Node.js)

User interface is friendly to users, with the user interface, users can handle our recommendation system without prior knowledge. So our project design a UI to attract new users.

We design the website based on the frame of Express and Reactjs. With the frame of express and reactjs, we are able to add new features easily in the future since reactjs is well support further maintenance.

Usage

Our tool is easy to handle, here is the detailed steps for usage.

1. Run rebuild_database/download_database.sh to download the dataset (Film Corpus 2.0 by UC Santa Cruz, https://nlds.soe.ucsc.edu/fc2) and parse the

dataset by AutoPhrase.

2. Run rebuild_database/init_db.sh to initialize MySQL.
3. Change directory to /express_reactjs, use command *npm start*, and then open a new session, change directory to /express_reactjs/ and then *npm start,* it will start a new webpage in the web browser.
4. To get the most similar films, enter either a block of scripts or several words from your favorite films at the right-upper corner of the page. You might want to find the scripts of your favorite movies at Springfield (https://www.springfieldspringfield.co.uk/movie_scripts.php) or the IMSDb database (http://www.imsdb.com/). It will also return the genres you might like based on the result of classification, and a couple of shared topic words of the films listed by the genres.
5. To search the database by keywords and figure out the most relevant movies and their scripts, you can put your keywords on the bottom of the page.

Individual Contributions

Luning Wang Implement quality words extraction, search engine and classification, build the user interface.

Muxia Yi Establish mathematical model to find the most suitable algorithms for SVM classification

Xinyu Tian Implement topic modeling, make the presentation, draft the documentation, and work as the coordinator.

References

[1] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, Jiawei Han, "Automated Phrase Mining from Massive Text Corpora", accepted by IEEE Transactions on Knowledge and Data Engineering, Feb. 2018.

[2] Jialu Liu*, Jingbo Shang*, Chi Wang, Xiang Ren and Jiawei Han, "Mining Quality Phrases from Massive Text Corpora", Proc. of 2015 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'15), Melbourne, Australia, May 2015. (* equally contributed, slides)