INFO440 PROJECT

Movie Recommendation Based on Tweets

Team 4: Harry Zhao, Carl Shen, Ziyan Di, Yi Pan

Background

As we all know, the covid-19 that began in 2020 has ravaged the world. In order to combat this emerging respiratory disease, governments of all countries have taken adequate measures to prevent the epidemic actively. When vaccines have not been widely vaccinated to form herd immunity, governments of all countries take the first measures to control people's travel, strictly control people at home, avoid going out as much as possible, and reduce contact between people. In this case, the fun of going out is lost, and people seek to watch movies at home to pass the time. At this time, the movie recommendation system becomes very meaningful. In the article "Predicting the Future with Social Media," the prediction of the future box office is significant, so we decided to make a movie recommendation project in this project.

Related Theory

Sentiment analysis

Sentiment analysis, also known as opinion mining, is a field of natural language processing (NPL), which is mainly used to identify and extract ideas from the text[1]. Sentiment analysis is very versatile, just as the online shopping application we usually use often recommends products to us or the movie recommendation system we chose in this project.

The basis of sentiment analysis is words, but the most basic text unit for expressing emotions is sentenced. Different words combine to express different sentiment tendencies. Therefore, it is necessary to analyze the sentiment tendencies of texts through sentences.

Generally, there are three types of sentiment analysis, namely positive, medium, and negative.

Recommendation System

The rapid growth of data collection has led to a new era of information. Data is being used to create more effective systems, and this is where recommendation systems come into play. Recommender systems are information filtering systems because they improve the quality of search results and provide more relevant items to the search item or the user's search history. There are three types of recommendation systems[2]:

Demographic filtering - They provide generalized recommendations to each user based on the popularity or genre of the movie. This system recommends the same movies to users with similar demographic characteristics. Since each user is different, this approach is considered to be too simple.

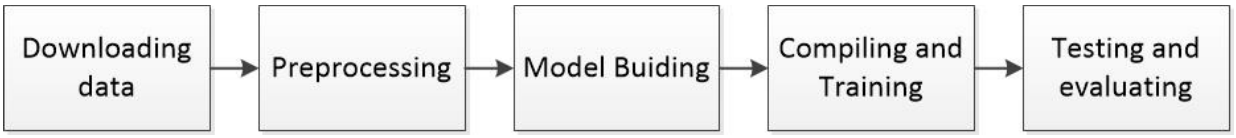
Content-based filtering - They suggest similar items based on specific items. This system uses project metadata, such as the movie genre, director, description, actors, etc., to make these recommendations. This approach has been used by a wide range of service providers and is well established.

Collaborative filtering - The system matches people with similar interests and provides recommendations based on that match. Collaborative filtering does not require the same item metadata as similar content-based systems. At the same time, the approach is still in the exploratory stage, and there is no mature and well-established solution[3].

Accomplishment

Sentiment analysis

In building the sentiment analysis model, the data I chose to build the model is the IMDB database, which is a dataset of movies' comments. This data set contains 50,000 comments with apparent biases, half of which are used as the training set and half as the test set.



The downloaded data has been processed and is an integer sequence. Each integer in the sequence represents a specific word, translated in the IMDB dictionary with a decode.

The dictionary of specific words can also be downloaded with code. This dictionary has more than 50,000 words, but we only use the first 10,000 because words that appear too low are not representative. These low-frequency words are not adequate when used in training data.

The data in the data set represent a comment, and the length of the comment is not the same. Therefore, we need to standardize the length to facilitate the training of the model. Using the word_index in the previous picture as a standard, the length of the data can be standardized. The part of the data that exceeds the standard is deleted, and the part of the insufficient length is filled.

Next is the process of building the model, which requires stacking layers in order to build a classifier.

The first layer is the embedding layer. This layer will look up each word-indexed embedding vector in the integer-encoded vocabulary.

The model learns these vectors when it is trained. These vectors add a dimension to the output array. The generated dimensions are: (batch, sequence, embedding).

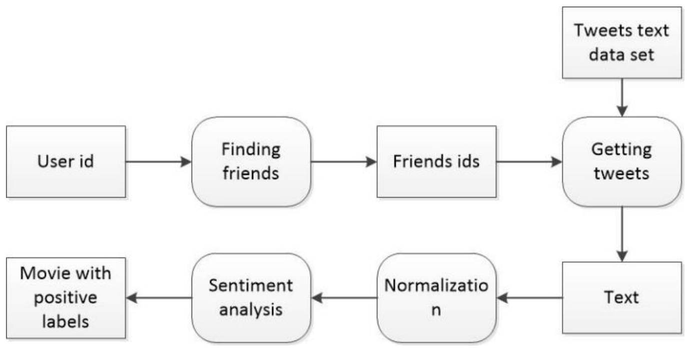
Next, a GlobalAveragePooling1D layer returns a fixed-length output vector for each sample by averaging the sequence dimensions. In this way, the model can handle inputs of various lengths in the simplest possible way.

The fixed-length output vector will be passed into a fully connected (Dense) layer (containing 16 hidden units).

The last layer is densely connected with a single output node. After applying the sigmoid activation function, the result is a floating-point value between 0 and 1, representing the probability or confidence level.

Recommendation System

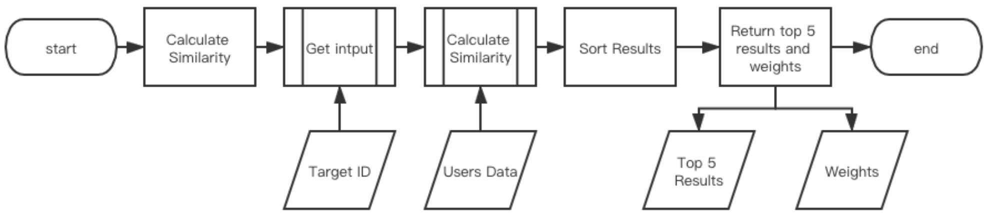
In the friendship subsystem, the main idea of this system is to find the tweet text in the tweet text data set created in data acquisition based on the ids of the friends found through a given user id. With the text found, sentiment analysis is applied to give the labels on the text towards the movie the text commented on. The label can be either positive or negative, meaning the user who commented on the movie is for or against the movie, respectively.



Firstly we break the comment text sentence into tokens to analyze it with the NLP module of Keras. The function's input is the frequency of each token of a comment text, and the output is the label of the comment. Firstly, For every normalized token in the comment text, we map the frequency of each token, then after transforming the data into the proper format, we use the method `keras.preprocessing.sequence.pad_sequences` to do sentiment analysis. We encapsulate the above procedure into the function called `senti_ana()`.

Then, we constructed the function called friend_recommand(). The function takes the user id as an argument. We are then using the get_friends() to find all friends' ids. After ensuring all the friends is in the text data set, for each friend, finding his/her comments in the data set and applying the sentiment analysis function to it will return the label of the text. Finally, the function returns the movies where the comment from friends is positive.

In the User similarity-based subsystem, we first obtained all the characteristics data of the user, including description, profile color, etc., and processed them separately. Sentiment analysis is performed on description, and the results of sentiment analysis are used as features. For color information, LableEncoder was used to convert them into a convenient form for manipulation. After that, the Euclidean distance between the user corresponding to our input and other users was calculated, and the five closest ones were selected. Then the movies that these five users feel POSITIVE are obtained and multiplied by the normalized percentage of similarity as the output value of this subsystem. The process is like following:



Weighting algorithm

In our project, the presence of two subsystems for recommendation produces two results. Our final results should be generated by considering the results of both subsystems. Therefore, we give each subsystem a weight of 0.5. For the friendship model, the movie he returns will directly carry a weight of 0.5. For the user similarity model, each result will also be multiplied by the normalized similarity as the weight again, according to the similarity from high to low, and finally returned. The calculation is shown in the following equation:

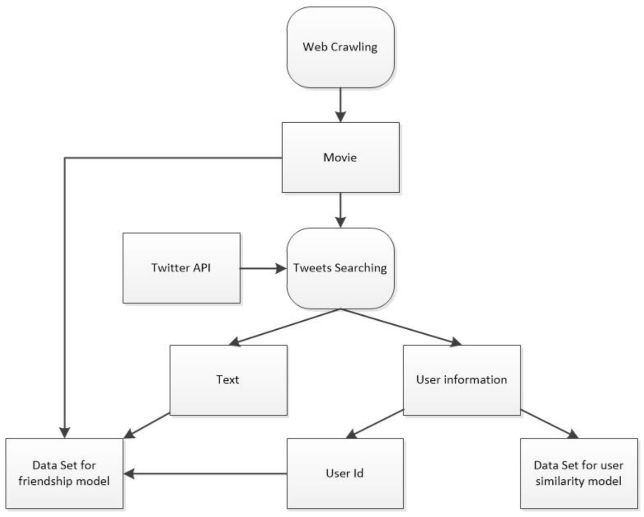
$$score = \omega_1 \times o_1 + \omega_2 \times s_2 \times o_2$$

Where ω_i is the weight, o_i is the result of model, and s_2 is the normalized similarity.

Experiment and Result

Data acquisition

The data we collected for the project is the movie comments from twitter. Based on that, we constructed a data acquisition and processing procedure.



ROTTEN TOMATOES is an American movie ranking website that mainly provides movie-related reviews, information, and news. This website has some pages that give top movies of a specific genre, such as Top 100 comedy movies, Top 100 action movies, etc. We choose several top pages to extract the movie names within its HTML. We did some research on the web HTML page, and we found that all the top pages in this website have the same structure to list the top movies, as can be seen below:

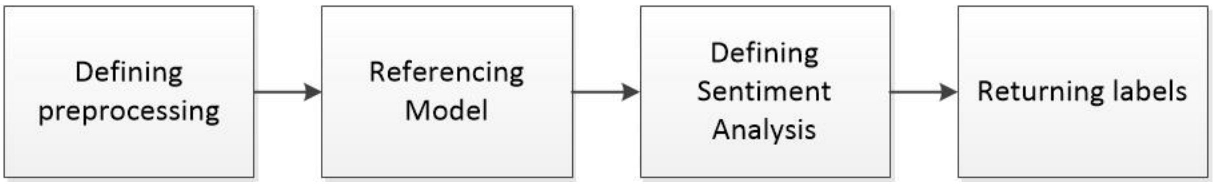
```
<div>/<div>/<section>/<div>/<table>/<td>/<a>movie name
```

In order to locate and extract movie names in html documents, we use xpath. xpath is a language for finding information in XML documents. As a result, with page structure and text extraction language, we apply a loop algorithm and collect 800+ movies.

Then, we used the code of twitter searching from module 'twitter', written by Prof. Shetty in Hackathon of week2 about TwitterMining to collect tweets. The essential method is `twitter.Twitter.search.tweets()`, which takes an argument as a topic to search relative tweets. With the methods, we can take the a movie as topic, and collect the tweets as movie comments with its statuses information including the text and commenting user's information.

FInally, with movies and their corresponding tweets information in hand, we build a data set for friendship model, the information(column) of it are user id, movie and comment text. Another data set contains the information of user, which is used for user similarity model.

Sentiment analysis



The main function of the sentiment analysis model is to input a piece of preprocessed film review data and then return to the sentiment prediction for this review. We chose the Keras method in Tensorflow because the content format of movie reviews is relatively more standard, which is suitable for processing the tweets data we obtain from the network after processing and sentiment analysis of tweets. Furthermore, because there are only two main movie review tendencies in the IMDB data set: positive and negative, neutral emotional tendencies were not considered in constructing the model.

In order to make it more convenient to use, let the function directly input the comment we want to test and then get the prediction result. We also need to use the defined norm function to preprocess the tweets. The norm function includes the functions of tokenization, word stemming, and removing punctuation for tweets.

Next, we used the sentiment analysis model that we have trained before, including data format processing and applying. The data we used must be an array with 256 dimensions.

We also get something useful from papers, especially the one that we selected: "World Cup 2014 in the Twitter World: A big data analysis of sentiments in U.S. sports fans' tweets."

According to this paper, we know that tweets sometimes can reflect users' thoughts and emotions. In this paper, people's emotions will be different when they were watching games. Even in some cases, the audience's mood will change as the situation of the game changes. For example, when the home team scores a goal or the opponent scores, users will post some tweets expressing their thoughts at the time. These tweets sometimes have some potential emotional tendencies. With these emotional tendencies, we can get some interesting information: the game's situation, the degree of importance of the game, and so on.

Because tweets are highly time-sensitive, users can frequently post tweets, so data can always be collected and analyzed in time, and analysts can continuously analyze the data to get the prediction results they want.

More importantly, the paper also mentions the preprocessing of raw tweets. The steps include noise cleaning, tokenization, capitalization conversion, stop-word removal, emoticons, stemming, and lemmatization.

Regarding these contents, we also used similar data preprocessing methods in the sentiment analysis model. In addition, we use natural language processing tools (NLTK) to process the data we want to process.

Recommendation System

We use userid 428549119 as input to show the result. After inputing this userid and comment data into friend_recommand() like `fri_movies = friend_recommand(428549119, comment)`, we can get a list named fri_movies, which is ['The Vast of Night', 'Paddington 2'].

Then, after we input the userid, comment and user data into user_sim_recommand() like `usr_movies, usr_weights = user_sim_recommand(428549119, comment, user)` we can get 2 lists named usr_movies and usr_weights, which are the name of movies and weights. In this case, usr_movies is [['Frozen'], ['Shaun of the Dead'], ['The Lunchbox'], ['The Lost Weekend'], ['Ernest & Célestine']], and usr_weights is [0.2541035146251795, 0.20089949269078483, 0.18767939367737027, 0.1835083500878237, 0.17380924891884164].

Then, we can combine these 2 subsystem by calling the recommand_movie(), and we will get a dictionary where the key is movie name and value is the score. In this case, the recommendation this system made is as shown like following:

```
No. 1 Movie name:    The Vast of Night      Weight: 0.500000
No. 2 Movie name:         Paddington 2      Weight: 0.500000
No. 3 Movie name:               Frozen      Weight: 0.127052
No. 4 Movie name:    Shaun of the Dead      Weight: 0.100450
No. 5 Movie name:         The Lunchbox      Weight: 0.093840
No. 6 Movie name:     The Lost Weekend      Weight: 0.091754
No. 7 Movie name:   Ernest & Célestine      Weight: 0.086905
```

Thus, we have the top 5 recommendations now, along with their score.

Conclusion

Project solution

In this project, Sentiment analysis implements sentiment analysis of tweets, adding positive or negative tags. The Friendship part analyzes the friends of the selected user to get corresponding movie recommendations. User similarity compares each user in the database with the selected user and gives the corresponding Euclidean distance similarity. The weighted result is the recommendation coefficient of the movie. The larger the coefficient, the more recommended to users.

Project results

In this project, we used 428,549,119 tweets. When a user is known to exist in our database, we will return a dictionary. The key of this dictionary is the name of the movie recommended to this user, and the value corresponding to the key is the recommendation coefficient of this movie. The larger the recommendation coefficient, the higher the recommendation degree.

Project advantages

We believe that the main advantage of this project is that these four parts can be put together to get good results. If we split them, we can also get the corresponding value from each part. For example, the separate user similarity part is calculated based on the similarity between users. Therefore, we think this project has excellent scalability.

Disadvantages of the project

1. The running speed is plodding. Because our data volume is enormous, and it is processed separately.

2. There are specific requirements for real-time calculation. Because our data comes from Twitter, these Twitter users generate tweets all the time, making it difficult for real-time calculations.

3. Only a small part of Twitter can be used for our experimental data. The number of Twitter users is vast, but we still only have a small number of users available for our projects with such a large base.

What has been learned in the project

In this project, we mainly learned the following two aspects:

First, in terms of team member cooperation. Each part is completed by different group members and communicates with each other to draw each person's needs and enhance everyone's collaboration ability.

Then, in terms of code. Be more proficient in using python for data analysis, and at the same time proficient in data processing under the platform of Twitter. This is a road that has not been traveled before and has a substantial reference value for us.

Prospects

In the future, we would like to explore more the use of social networks in recommender systems. In our project, we used friendship but only explored a deep layer of social networks. As social networks continue to expand, people are in one group after another, with differences between groups and similarities. In the future, we hope to be able to make recommendations through this.

Reference

[1] Feldman, R. (2013). Techniques and applications for sentiment analysis. Communications of the ACM, 56(4), 82-89.

[2] Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S., & Stettinger, M. (2014). Basic approaches in recommendation systems. In Recommendation Systems in Software Engineering (pp. 15-37). Springer, Berlin, Heidelberg.

[3] Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R., & García-Sánchez, F. (2012). Social knowledge-based recommender system. Application to the movies domain. Expert Systems with applications, 39(12), 10990-11000.

And all the reading from this course.