Free Topics - Movie Similarity Analyzer

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members
   Chenyao Yin - chenyao7
   Zihan Qiu - zihanq3
   Yunchang Pang - yp9
   Shuhao Kang - shuhaok2

2. What is your free topic? Please give a detailed description. What is the task? Why is it important or interesting? What is your planned approach? What tools, systems or datasets are involved? What is the expected outcome? How are you going to evaluate your work?

   We want to build an analyzer for movies by checking the syntagmatic similarity between their plot overviews.We will create an algorithm that evaluates the similarity between two movies with their syntagmatic similarities, genres, release year, and many other factors. It would be interesting to find similarities between movies based on factors other than just genres and actors. And we are expecting to explore certain patterns in plots from the recent “Hollywood streamline products”. We will scrape movie info from IMDB and gather all the data we need for checking similarities and store them in our databases. When a user checks the similarity between two movies, our frontend will request our backend where all the pre-processed data will be used to calculate the similarity between those movies.

   We are going to evaluate the accuracy of our algorithm by gathering user experience of our website. We will let our user judge if the similarities between the movies are reasonable.

3. Which programming language do you plan to use?
   Python,
   JavaScript(React)

4. Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

   Python crawler 5h (Scrapy)
   Obtaining the plot summary and many other factors that we take into consideration when comparing movies.

Plot synopsis parser & preprocessing 10h
Preprocess the scraper content into plain text and format that's more approachable by the following algorithm, i.e. building vocabulary around it

Corpus Vocabulary Builder 30h
The python functions that implement PLSA algorithm, to find a probabilistic model with latent or hidden topics that can generate the data which we observed in the in document-term matrix fetched

API backend server 8h
The socket server that provides access to the database from our frontend website.

MongoDB database 8h
The DB that stores completed word clusters and frequencies.

React Frontend & Data Visualization 20h
A website that allows visualization of two movie similarities based on syntagmatic similarities from movie plots.

Server Host 8h

- Your documented source code and main results.
- Self-evaluation. Have you completed what you have planned? Have you got the expected outcome? If not, discuss why.
- A demo that shows your code can actually run and generate the desired results. If there is a training process involved, you don't need to show that process during the demo. If your code takes too long to run, try to optimize it, or write some intermediate results (e.g. inverted index, trained model parameters, etc.) to disk beforehand.