CS 410 Final Project Proposal

Team “Bae Area”

Free Topic

Improved Media Content Search

Team Members

Brayden Turner - brturne2@illinois.edu - *Captain*
Joshua M. Smith - jms28@illinois.edu

Description

We’ve all tried to remember a song or movie but all we have is a rough description of what the song or movie is about. Current search tools for music and movies are limited to metadata such as titles, artist, actors, but not the general sentiment of lyrics†e content, and using the lyrics and movie synopsis, return a ranked list of content that fits the description given. This can also be extended to add tags to content like “Song from Super Bowl 40 Halftime” or “Rolling Stone top 10 movies list” to enhance descriptive search.

We plan to use Python libraries (*lyricsgenius* and *imdbpy*) to retrieve information from Genius and IMDB (respectively) to build our indexes and run our models on. We expect to be able to give a description of a media entity and retrieve back a ranked list of entities matching the description. We plan to take a number of entities with notable plots (e.g. Lord of the Rings, Star Wars, etc.) and build queries we know should return them. This will give us a test set of queries + entities that we expect. Using that we can get precision and recall (are all 6 of the Lord of the Rings Movies in the opt 10?).

Resources and Tools

- Lyrics / song data - https://pypi.org/project/lyricsgenius/
- Movie / Synopsis data - https://imdbpy.github.io
- MeTA (inverted index)

Programming Language

Python

Task Breakdown

For two members, we estimate 2*20 = 40 hours of work.

We anticipate the following tasks:

- Set up work environment and infrastructure (3 hours)
- Build a dataset based on music or movie databases using available Python libraries and APIs (5 hours)
- Build a set of relevance judgements and tests for evaluation (4 hours)
- Construct an inverted index to use for searching the dataset (5 hours)
- Create a search engine over the dataset (5 hours)
- Implement the model, e.g. Naive Bayes, PLSA (10-15 hours)
- Evaluate search performance (5 hours)
- Write our final report with conclusions (5 hours)
- If time allows: create web frontend for application (5 hours)