CS 410 Project Proposal, Fall 2023

Team Info

- Manuel Suarez Lunar | manuel6@illinois.edu
- Wei-Lun (Will) Tsai | wltsai2@illinois.edu --> team captain

Topic Overview

Our team project will be focused on the development of a unique search engine / text retrieval tool specifically tailored for literature and quote enthusiasts. Leveraging the vast repository of quotes available on Goodreads.com, our text retrieval tool will use a web crawler to sift through the website's quotes content. Unlike traditional search engines, ours will be sentiment-centric; users will input a particular sentiment or emotion, and our system will return a ranked list of famous quotes resonating with that sentiment. Additionally, in parallel to the core functionality of the search engine, we would also like to introduce an add-on feature enhancing the user's experience. By capitalizing on the same sentiment input, this feature will produce a ranked list of authors whose body of work predominantly aligns with the specified emotion or sentiment. This will not only allow users to discover quotes but also introduce them to authors who resonate with their current feelings, enabling a deeper exploration of literature in tune with their emotional state. And finally, we would also build a web interface where the user can submit the inputs for both search functionalities.

We found this topic to be interesting and useful because we think our search engine can help users discover and connect with literary quotes, providing a more intuitive, focused and emotion-driven user experience than current mainstream Google or Bing can provide. Also, with the added functionality of searching and ranking author names, our search engine can be used by parents looking for name ideas for their children.

Plan and Approach

Our planned approach will consist of four main components: web crawler, index of text data, text retrieval backend, and web interface. The web crawler will be responsible for collecting text data of famous quotes attributed to various authors or speakers from Goodreads.com. The index of text data will be responsible for storing the text data in a way that may be efficiently searched for text retrieval and will include Term Frequency and Inverse Document Frequency representations. The text retrieval backend will be responsible for taking in user-supplied text as input, determining the sentiment of the input, and then returning a ranked list of quotes and respective authors that resonate with that sentiment. Lastly, the web interface will be responsible for allowing the user to submit their desired input and then view a ranked list of quotes and authors based on their sentiment, allowing users to supply their feedback through like/dislike buttons.

The expected outcome is a web application that allows users to input a sentiment and view a ranked list of quotes and authors that resonate with that sentiment. The web application will be hosted on a server and will be accessible to users via a web browser.

We plan to use the following evaluation methods for our text retrieval tool:

- **F1-Score**: combines both precision and recall to provide a unified view of search quality

- **Mean Average Precision (MAP):** averages the precision scores after each relevant quote is retrieved
- **Normalized Discounted Cumulative Gain (nDCG)**: evaluates the ranking quality of search results, taking into account the position and relevance of each result
- **Click-Through Rate (CTR) / Relevance Feedback**: we'll share our tool with 10+ test users and ask them to click on the results that are relevant to them. Alternatively, we can create a survey to capture explicit user feedback on the results quality

Technology and Implementation

Primary datasets include the famous quotes raw text data collected by our crawler, indexed searchable quotes text data, and sentiment analysis data from Scikit-Learn. For the web crawler, we will use the Python library Scrapy, a popular open-source web scraping tool. With the help of the metapy library, we will build our own bag of words text representations, TF, and IDF of the famous quotes data collected by our crawler. For the text retrieval backend, we will leverage the Scikit-Learn Python library to perform sentiment analysis on the quotes. For the user frontend, we will use the Python library Flask to build the web application and leverage a framework like AngularJS to create the user interface. Finally, as a stretch goal, we will host our web application on the cloud, using a service like Azure Web Apps.

We plan to use Python for our crawler and text retrieval (backend). Later, we plan to use JavaScript for the web application and user interface (frontend).

Milestones and Task Estimates

We anticipate of our project deliverables to take at least 50 hours combined with two team members, consisting of the following main tasks:

- Web crawler to collect famous quotes data (10 hours)
- Index of text data, including TF and IDF (10 hours)
- Text retrieval backend with sentiment analysis (10 hours)
- Web application and interface (10 hours)
- Feedback capture and evaluation component (5 hours)
- Hosted application on Azure (5 hours)