SSD Term Project for Monsoon’22

Topic - Retrieval Engine

Team Members :-

- Rahul Padhy (2022201003)
- Ujjwal Prakash (2022202009)
- Gunank Singh Jakhar (2022201057)
- Venkata G Srikiran Pulipaka (2022204004)

Team Guide :- **Dr. Charu Sharma**

Objective of a Retrieval Engine

A retrieval engine is used to extract answers to queries using contextual information from various sources. In this project, sources used are various search engines such as Bing, ask.com and information about text, news, movies, images and videos is returned as answer to user queries. Retrieving results fall under the category of machine learning too - this isn’t a classification problem per se (since no use of labels is there), but results are to be fetched in a particular order using a set of filters.

Project Design

- Five categories of search are provided by this Retrieval Engine - text based global search, image search, news, movies and videos.

- For **text based global search module**, no restriction is there on the basis of content to be searched about.
    - Scraping has been done from ask.com (with the actual intention of not being dependent upon more modern search engines such as Google, DuckDuckGo, Bing, etc.) and perform ranking on the fly.
    - The text search results shown have the following components - title of webpage (with embedded link), url and a short description.
    - This module uses TFIDF (Term Frequency - Inverse Document Frequency) algorithm for ranking the retrieved query results.

    - tf-idf is a weighting scheme that assigns each term in a document a weight based on its term frequency (tf) and inverse document frequency (idf).
    - The terms with higher weight scores are considered to be more important.

- For **image based search module**, no restriction is there on the type of image to be searched.
    - Images are retrieved from Google Images and are shown in the order in which they are received.
    - Only the images are shown here, with fixed height and width.

- For **news based search module**, the user enters their choice through two dropdown menus - Country and Sub-Topics.
    - **Allowable Country values ->** "Australia", "Botswana", "Canada", "Ethiopia", "Ghana", "India", "Indonesia", "Ireland", "Israel", "Kenya", "Latvia", "Malaysia", "Namibia", "New Zealand", "Nigeria", "Pakistan", "Philippines", "Singapore", "South Africa", "Tanzania", "Uganda", "United Kingdom", "United States", "Zimbabwe", "Czech Republic", "Germany", "Austria", 'Switzerland', 'Argentina', 'Chile', 'Colombia', 'Cuba', 'Mexico', 'Peru', 'Venezuela', 'Belgium ', 'France', 'Morocco', 'Senegal', 'Italy', 'Lithuania', 'Hungary', 'Netherlands', 'Norway', 'Poland', 'Brazil', 'Portugal', 'Romania', 'Slovakia', 'Slovenia', 'Sweden', 'Vietnam', 'Turkey', 'Greece', 'Bulgaria', 'Russia', 'Ukraine ', 'Serbia', 'United Arab Emirates', 'Saudi Arabia', 'Lebanon', 'Egypt', 'Bangladesh', 'Thailand', 'China', 'Taiwan', 'Hong Kong', 'Japan', 'Republic of Korea'.
    - **Allowable Subtopic Values ->** "Business", "Technology", "Entertainment", "Sports", "Science", "Health".
    - The news results shown have the following components - Title of the news article (with the corresponding link embedded), news article summary.
    - Summarization has been provided for each news article and the summaries have been generated taking the help of nltk and spacy modules in Python.
    - The method relies on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary.

- For **video based search module**, the user has been given the full freedom to search for videos based upon any topic.
    - Scraping has been done from Bing Videos, which contains videos from multiple sources such as YouTube, FaceBook, Instagram, DailyMotion, etc.
    - The video results shown have the following components - Title of the video (with the corresponding link embedded), channel-name, channel-platform, date-uploaded and number of views.
    - Ranking has been done on the basis of number of views for a particular video.

- For the **movie based search module,** the user is given options to view their choice of movies from the dropdown menu based upon Genres.
    - **Allowable Genre values** -> 'Comedy', 'Sci-fi', 'Horror', 'Romance', 'Action', 'Thriller', 'Drama', 'Mystery', 'Crime', 'Animation', 'Adventure', 'Fantasy', 'Superhero'.
    - Scraping has been done from imdb.com.
    - Ranking has been done on the basis of imdb_ratings (from audience) and metascores (from critics).
    - The components displayed are :- movie name (with the corresponding embedded link), year of release, movie rating (age), list of genres corresponding to that particular movie and movie run-time (in minutes).

Tech Stack used for the project

- Backend has been done in Django (Python).
- Frontend has been implemented using HTML / CSS / JS.
- External Python libraries such as BeautifulSoup, scikit-learn, nltk, spacy and Newspaper have been used for scraping, TFIDF, summarization and text processing respectively.
- The left footer contains links to some commonly visited websites such as LinkedIn, Twitter, Instagram, Github and Reddit.
- The right footer contains link to the project’s source code.

Limitations

- Latency is observed in the case of scraping.
- Text processing and summarization take a bit of time to execute since the entire content of web page needs to be processed on the backend.

Link to Source Code

https://github.com/JimHalpert26/T5_Retrieval_Engine