**Enhancing Search Engine Relevance for Video Subtitles**

**Background:** In this era of technology, search engines play a vital role in serving user queries seamlessly. From brands of tech to beauty each website has their own search engines. In the same way, when it comes to video subtitles, a proper search option is required to get the relevant results in just one word query.

**Objective:** This project completely focusses on developing efficient search engine based on user query, with specific emphasis on subtitle content. The objective also involves application of natural language processing and machine or deep learning models to enhance the accuracy of search results.

**About the dataset:** The dataset provided is a huge collection of 82498 subtitle files that is encoded in Latin format and compressed in a database file. Each subtitle files had its unique id and subtitle content with dialogues number.

**Chunking:** This process involves breaking down the huge collection of files into chunks for easy processing and to avoid memory error. Since the data is very huge only 30% of it is considered for analysis and after applying chunking and decoding of subtitle files 7 csv files are created. The csv files contain columns that represent the id, title of subtitle file, file encoded content and file decoded content of each subtitle files.

**Libraries Used:**

* Database connector – sqlite
* Extractor – zipfile, io (for decoding)
* Directory - os
* Dataframe – pandas, numpy
* Cleaning – re
* Visualization – wordcloud
* Vectorization – torch, BertTokenizer and BertModel from transformers, scikit-learn, scipy-sparse.
* Embeddings – chromadb, Ipython.

**Data Cleaning:** Following steps are involved in data cleaning-

* After extracting the data from database file and decoding, it is saved in a csv file that is then chunked to seven chunks for easy processing.
* Only 40% of data is used in model development.
* Timestamp removal, unnecessary character (like /r/n/d) removal, unnecessary special characters removal and dialogue number removals are performed using regex expression.

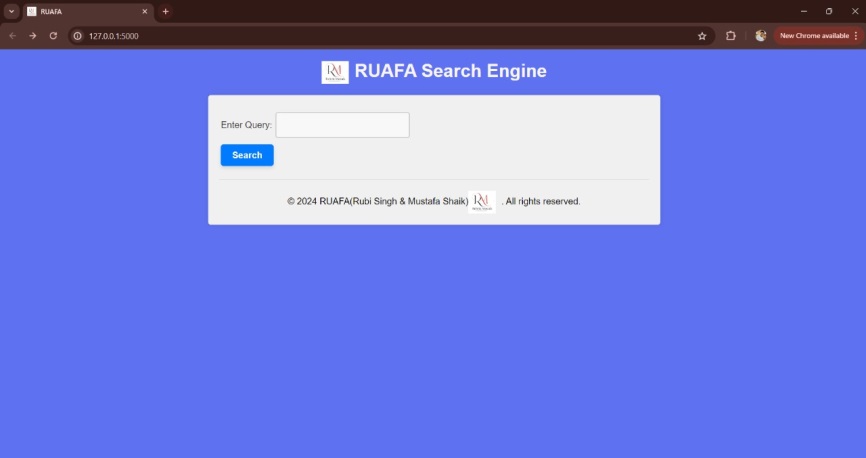
**Vectorization:** This process involves conversion of text into numerical embeddings for easy searching. Two types of vectorization techniques are used in this project one with transformers and another one with tfidf and count vectorizer. The idea is to observe the quality and issues of vectorizing with both models. Tfidf and count vectorizer techniques increase the dimension of data and the matrix generated by it is highly sparse. Transformers generate low dimension low sparse embeddings.

**Chromadb:** It is an open source database platform where the numerical embeddings with the content can be stored. Search operation can be performed on it and based on the query it will return the most similar documents. Any type of distance methods can be used like Euclidean, Minkowski but the most common method used for text searching is cosine similarity.

**Flask Based APP:** Flask backend is used for the search engine development. The interface consists of query box where the user can pass the input or keyword. The query is then converted to embeddings then using cosine similarity top 10 subtitle’s title and ids are displayed. Each subtitle displayed has a unique button. If the user wishes to see the whole content then the button can be clicked. There is also buttons at the end of each row which will open the source website of the subtitles.

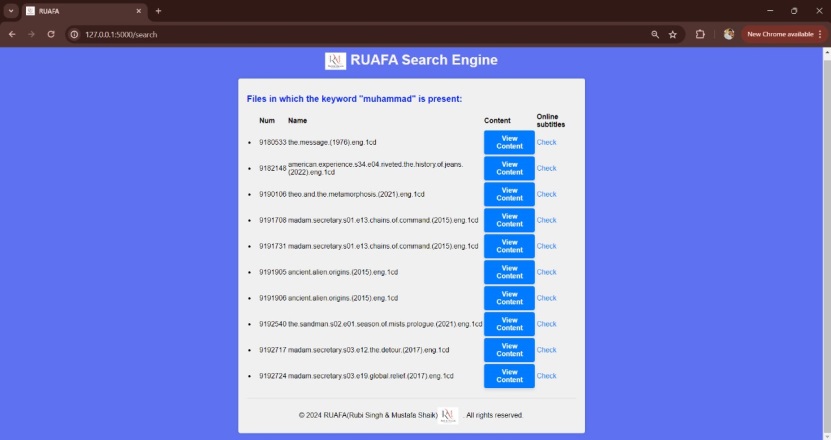
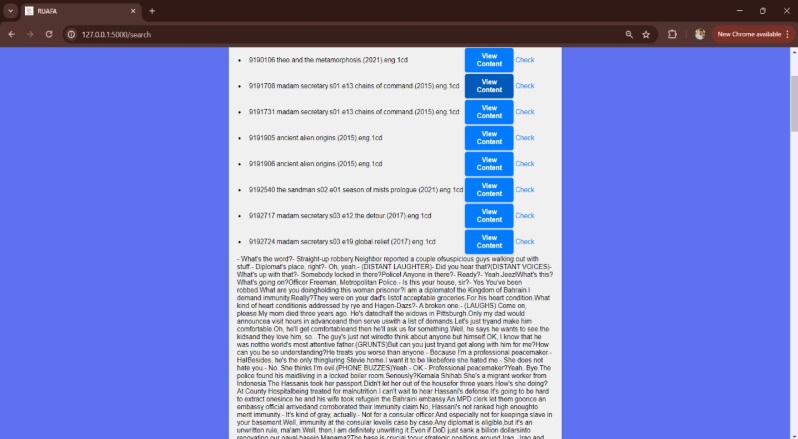
Following images shows the working of Search Engine –

**Text area to input query**



**Button for searching**

**Search engine Interface**



**Content of the file clicked**

**Top 10 relevant Subtitle title**

**Conclusion:** The project aims to highlight the pros and cons of keyword and semantic based searching. It signifies the importance of semantic search as it is not just based on exact matchings but focusses on the meaning of the words provided. Depending on the limited resources available this is a very basic search engine developed. Further scope for advancement with more data and more training to models can be researched.

Contributors: Mustafa Shaik and Rubi Singh.