Information Retrieval

Mini Project Report

Movie Plots Retrieval

Group - P07

Group Members

- Vaidya Rugvedh, S20190010185
- A Pranay, S20190010001
- A Likhith Bharadwaj, S20190010008
- Veduruparthi Sai Bhaskar, S20190010188

Date of submission: 17 - 12 - 2021

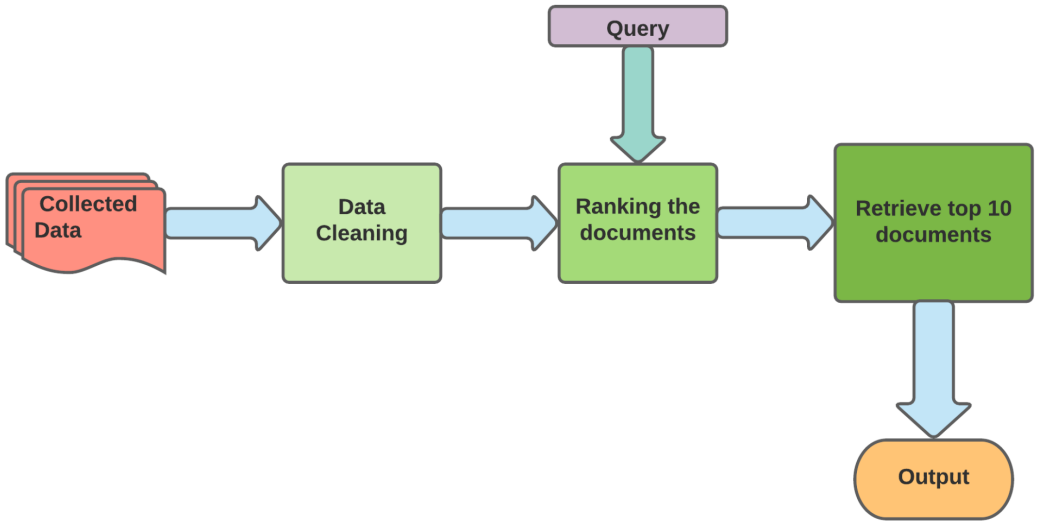
Problem Statement :

To build a search engine on Movie Plots

Project Description :

- The main goal of an ir project is to develop a model for retrieving the information from the repositories of documents.
- A model of information retrieval predicts and explains what a user will find in relevance to the given query. IR model is basically a pattern pattern that defines the above-mentioned aspects of retrieval procedure and consists of the following

Flow diagram:



Task-1:

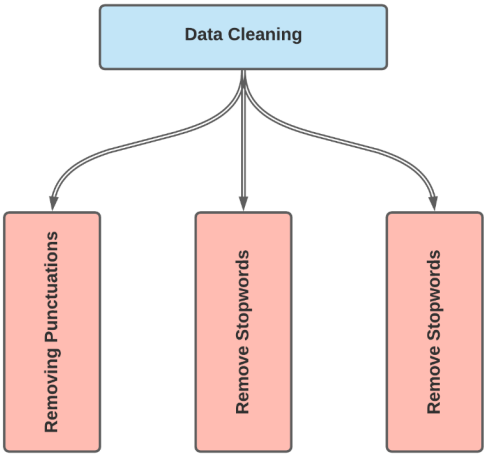
Data collection:

We collected a dataset from Kaggle on Movie plots(Wikipedia Movie Plots). The data is in the form of csv format.

You can also download the data in the csv format using following link
wiki_movie_plots_deduped.csv

Task-2

Data cleaning:



- Removing Special characters
  - The special characters [!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~] are removed using a function text_cleaner.

- Removing Stop Words
  - The stopwords are removed using nltk packages which are in the function remove_stopwords.
- Tokenization
  - The data is tokenized using the nltk.word_tokenize default function in the preprocess.

Task-3

INDEXING

❖ **Inverted Index Construction**

We have built a slightly modified version of Inverted Index which comprises of

{

Term

Doc.Frequency

Posting list

We are appending idf score to Inverting Index as it is unique for each term

}

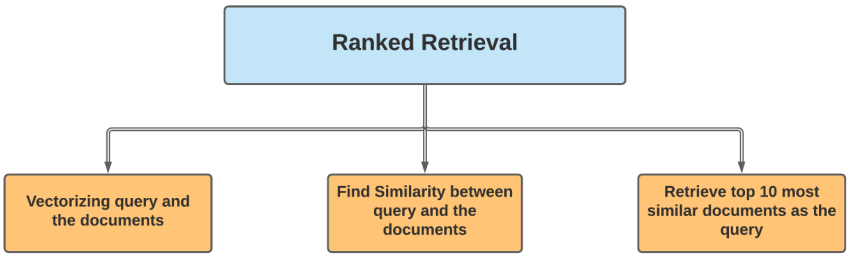
Example:

```
yachting': {'count': 3,
          'posting_list': [
     {'count': 1, 'docID': 436},
                         {'count': 1, 'docID': 641},
                         {'count': 1, 'docID': 957}
     ]
}
```

➢ The inverted indices are stored in a dictionary using **add_term_to_inverted_index** function.

Task-4

Vector space model



Vectorization:

Documents and queries are represented as vectors in the index terms space

Term weighting for documents

Term-frequency(tf)

- The number of times a word w appeared in a document d.

$$tf_{i,j} = 1 + log_2(freq_{i,j}),\ if\ freq_{i,j} \geq 1;\ else\ 0$$

Inverse document frequency(idf)

- This is another form of document frequency weighting and often called idf weighting weighting or inverse inverse document frequency weighting.
- The term's scarcity across the collection is a measure of its importance and importance is inversely inversely proportional to frequency of occurrence.

$$idf_i = log(\frac{N}{n_i})$$

Here, N = documents in the collection

n_i= documents containing term ti

The idf is calculated using calc_idf function

Tf-idf score is calculated and saved in tfidf_dict

Task-5

Query Formulation and Processing

- We have processed our query to remove stop words and special characters because they are not going to be used in term weighting

Term weighting for Query

- The term weighting for a query is calculated and stored in query_dict.
- The tf-idf score for the score is stored and saved in the dictionary.

Task-6

Ranking

Final Ranking using Cosine similarity

Cosine similarity:

- The cosine similarity is found between the query and each document is stored and used for ranking the documents.

It is nothing but the cosine of the angle between the query vector and document vector.

$$C.\ S = \frac{q^T d_j}{||q|| ||d_j||}$$

Ranking of documents:

The retrieved documents are arranged according to their cosine similarity and the top 10 documents are displayed.

Task-7

Evaluation

We have taken input from users for evaluation. The user is asked to type 1 for relevant and 0 for non relevant

Precision

$$p = \frac{\#(relevant\ retrieved)}{\#(retrieved)}$$

$$= \frac{tp}{tp+fp}$$

Recall

$$r = \frac{\#(relevant\ retrieved)}{\#(relevant)}$$

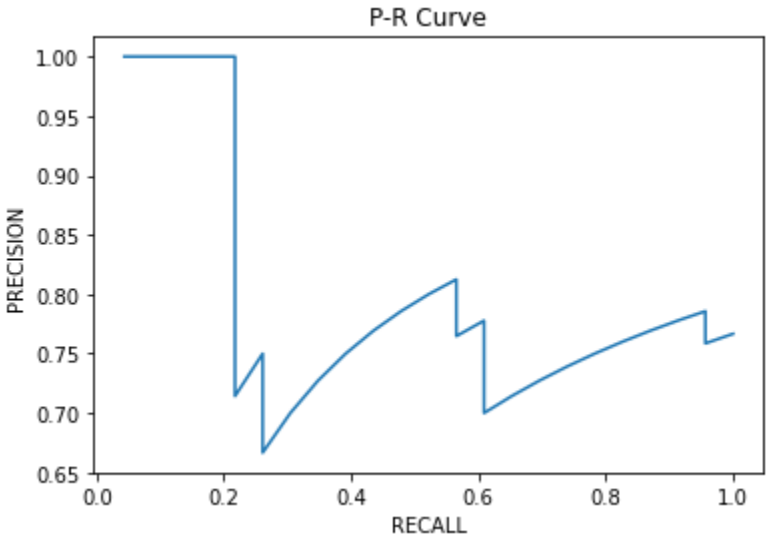
$$= \frac{tp}{tp+fn}$$

If We have given the query as “Young Women”

The retrieved documents are as follows

	Release Year	Title	Origin/Ethnicity	Director	Cast	Genre	Plot
87	1914	Mabel's Blunder	American	Mabel Normand	Mabel Normand, Charly Chase, Al St. John	comedy	Mabel's Blunder tells the tale of a young woman who is secretly engaged to the boss's son.[1] The young man's sister comes to visit at their office, and a jealous Mabel, not knowing who the visiting woman is, dresses up as a (male) chauffeur to spy on them.
63	1914	Between Showers	American	Henry Lehrman	Charlie Chaplin, Ford Sterling, Chester Conklin		Chaplin and Sterling play two young men, Masher and Rival Masher, who fight over the chance to help a young woman (Clifton) cross a muddy street. Sterling first sees the woman trying to cross and offers her an umbrella he stole from a policeman. He asks her to wait for him as he goes to get something to help her. Chaplin comes along and offers the woman to help her cross the street as well and wait for his return. While Sterling and Chaplin go to get logs, a policeman (Conklin) lifts the woman across the street. When Sterling returns with the log, he is indignant that the woman did not wait for him to come back to help her cross the muddy street and demands the umbrella back. When the woman refuses, they engage in a fight which eventually involves Chaplin.
42	1911	Sweet Memories	American	Thomas H. Ince	Mary Pickford, King Baggot	drama	Polly Biblett (Mary Pickford), a young lady, tells her grandmother Lettie about her new boyfriend. The news provokes the elderly woman to reminisce about her own sweetheart, long time before. The touching sequence expresses the power of lives going on, the older woman aging as her grandchildren grow and knowing they will soon have children of their own.
64	1914	The Avenging	American	D. W.	Henry B. Walthall,	drama,	A young man (Henry B. Walthall) falls in love with a beautiful woman (Blanche Sweet), but is prevented by his uncle (Spottiswoode Aitken) from pursuing her. Tormented by visions of death and suffering and deciding that murder is the way of things, the young man kills his uncle and builds a wall to hide the body.\nThe young man's torment continues, this time caused by

The P-R curve is as follows



Non Trivial task 1

For Non trivial task we are doing filtering by year where we are retrieving all the documents which are released in the same year

Non Trivial task 2

For Non trivial task we are doing filtering by Genre where we are retrieving all the documents which are released in the same year

WORK DISTRIBUTION

Rugvedh →Cosine similarity for ranking, Non Trivial task2

Likhith →TF-IDF Score for Documents, Query,Query processing, Non TrivialTask1

Pranay →Preprocessing, Construction of inverted index, Non TrivialTask1

Bhaskar →Evaluation, Non Trivial task2