NEWS SEARCH ENGINE - Design Document

Made By -

1. MIR AMEEN MOHIDEEN - 2018A7PS0487H
2. MD WAQUAR WASIF - 2018A7PS0254H
3. TULAIB AHMED ABDULLAH - 2018A7PS0272H

We first start preprocessing the collection of documents. The process contains two major preprocessing.

1. **Tokenization** : Tokenization is the process breaking complex data like paragraphs into simple units called tokens.
2. **Removal of stop words and punctuation :** We remove stop words and punctuations which are of no importance to the search engine.
3. **Normalization** : In TF-IDF, normalization is generally used in two ways: first, to prevent bias in term frequency from terms in shorter or longer documents; second, to calculate each term's idf value (inverse document frequency). Once idf_i is calculated, tf-idf is tf multiplied by idf

TF-IDF (Term Frequency-Inverse Document Frequency):

Tf-idf stands for *term frequency-inverse document frequency*, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

In our project we first calculate inverse document frequency for all the documents and then we calculate tf-idf for all the documents. We have defined a utility function to find tf-idf.

We calculate tf and idf as taught in class which is.

- **TF: Term Frequency**:- which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more time in long documents than shorter ones. Thus, we take the log of term frequency as a way of normalization:

  TF(t) = 1+log (Number of times term t appears in a document).

- **IDF: Inverse Document Frequency**:- which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

  IDF(t) = log(Total number of documents / Number of documents with term t in it).

- **Cosine Similarity**:- Finds the amount of similarity between 2 vectors.

  cos(q,d) = q.d/(|q||d|)

After finding tf-idf for all the documents preprocessing is done. We then take an input query and find the cosine similarity between the query and the document.

After everything is done we then sort the top 10 cosine similarities and display the corresponding results.

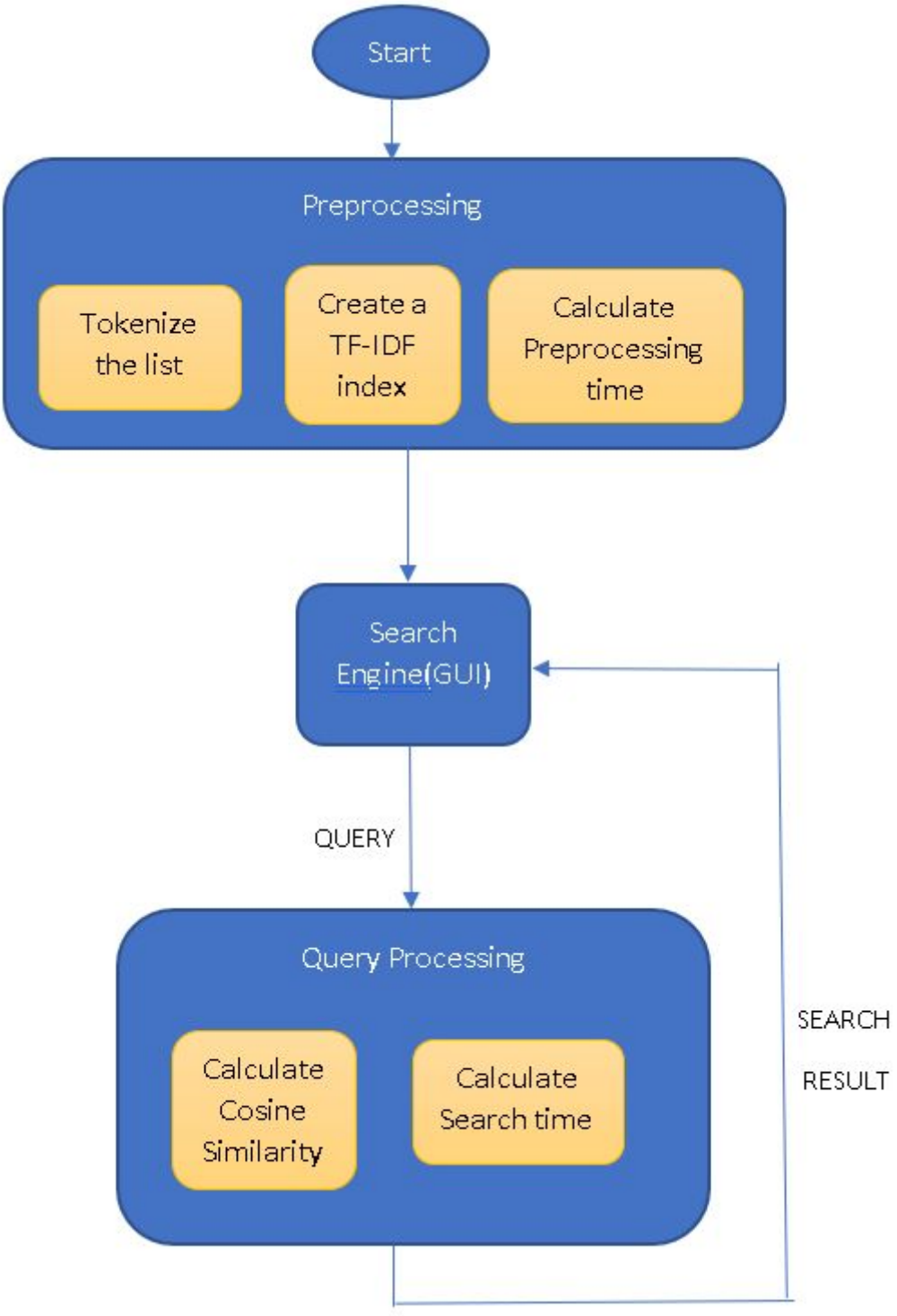
Data Structures:

Data structures are a way of organizing and storing data so that they can be accessed and worked with efficiently. They define the relationship between the data, and the operations that can be performed on the data.

We have used few data structures to make the processes easy. Which are as follows

1. **Data frame** : A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. In our code we have read the dataset using pandas and stored it in a dataframe.
2. **List** : We used List Data Structure to store news_sum, query_doc_sim(cosine similarities) and stop words.
3. **List of lists** : It was used to store tokenized strings(nt)
4. **Dictionary** : It was used to store idf values and indexes of unique words in the collection.
5. **numpy.array** : It is used to store idf ,tf and tf_idf values for all the documents.

Flowchart:

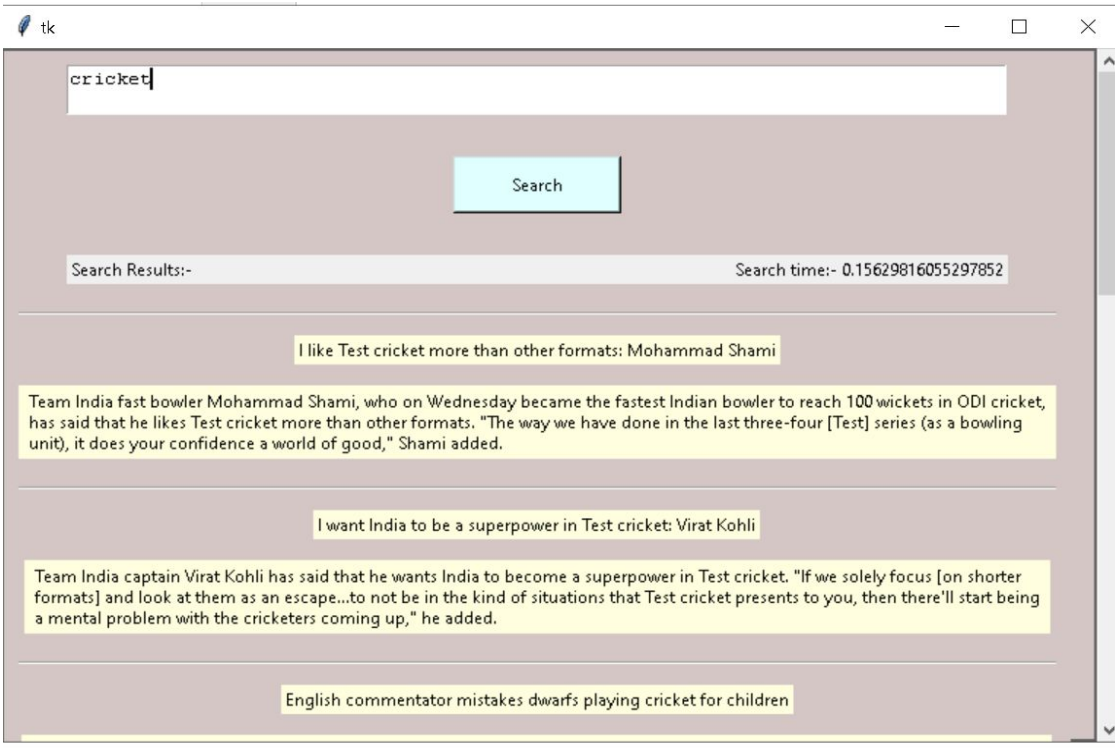

Start
Preprocessing
Tokenize the list
Create a TF-IDF index
Calculate Preprocessing time
Search Engine(GUI)
QUERY
Query Processing
Calculate Cosine Similarity
Calculate Search time
SEARCH
RESULT

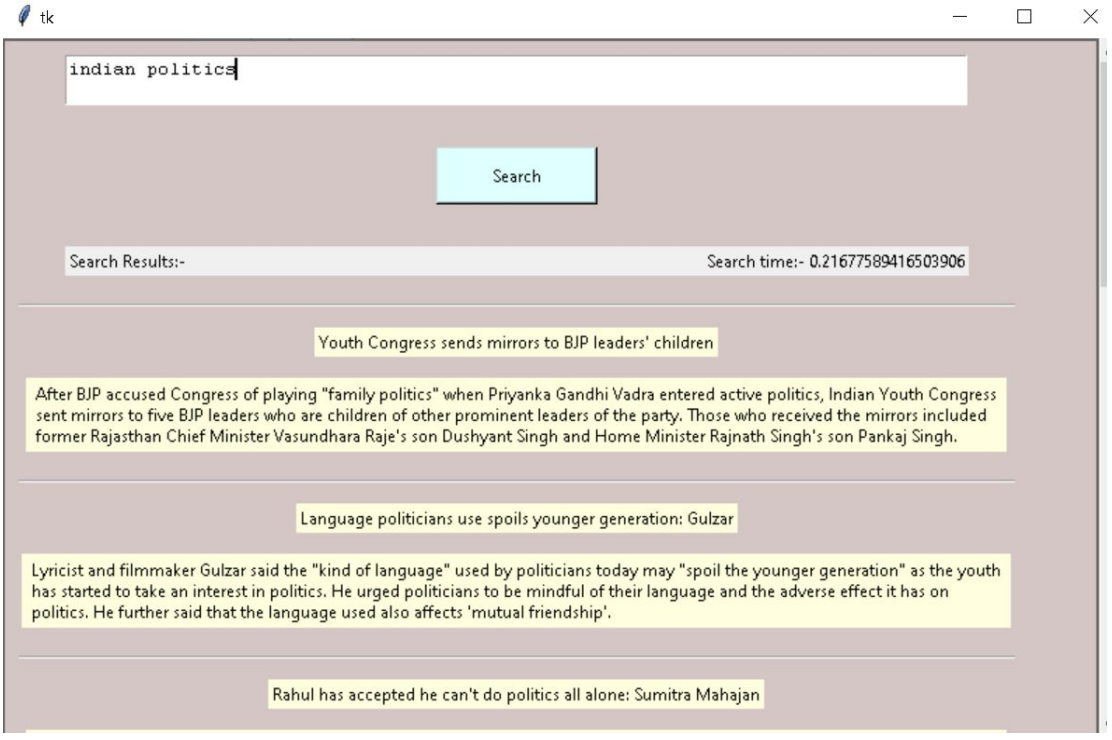
ScreenShots:

Preprocessing Time: ~ 5 seconds.

```
C:\Users\tulaib pc\Downloads\ir_final_1>python gui.py
(2000, 15191)
Preprocessing done!
Time to preprocess : --- 4.818526268005371 seconds ---
```

Search Time: ~ 0.20 seconds


tk
cricket
Search
Search Results:-
Search time:- 0.15629816055297852
I like Test cricket more than other formats: Mohammad Shami
Team India fast bowler Mohammad Shami, who on Wednesday became the fastest Indian bowler to reach 100 wickets in ODI cricket, has said that he likes Test cricket more than other formats. "The way we have done in the last three-four [Test] series (as a bowling unit), it does your confidence a world of good," Shami added.
I want India to be a superpower in Test cricket: Virat Kohli
Team India captain Virat Kohli has said that he wants India to become a superpower in Test cricket. "If we solely focus [on shorter formats] and look at them as an escape...to not be in the kind of situations that Test cricket presents to you, then there'll start being a mental problem with the cricketers coming up," he added.
English commentator mistakes dwarfs playing cricket for children


tk
indian politics
Search
Search Results:-
Search time:- 0.21677589416503906
Youth Congress sends mirrors to BJP leaders' children
After BJP accused Congress of playing "family politics" when Priyanka Gandhi Vadra entered active politics, Indian Youth Congress sent mirrors to five BJP leaders who are children of other prominent leaders of the party. Those who received the mirrors included former Rajasthan Chief Minister Vasundhara Raje's son Dushyant Singh and Home Minister Rajnath Singh's son Pankaj Singh.
Language politicians use spoils younger generation: Gulzar
Lyricist and filmmaker Gulzar said the "kind of language" used by politicians today may "spoil the younger generation" as the youth has started to take an interest in politics. He urged politicians to be mindful of their language and the adverse effect it has on politics. He further said that the language used also affects 'mutual friendship'.
Rahul has accepted he can't do politics all alone: Sumitra Mahajan