Information Retrieval Systems

Submitted By

Smeet Kothari

Under Guidance Of

Professor Kal Bugrara

1. **Calculating TFIDF(Term Frequency Inverse Document Frequency):**

To calculate TFIDF I have defined four entities namely Number of Occurrences, Total Terms in Document, Total number of Documents and Number of Documents with Term.

To calculate TF:

tf = No of Occurrences / Total number in Documents ;

To calculate IDF:

idf = Math.log10(Total number of Documents) / Number Of Documents with Term;

and then (tf \* idf \* 1000) is returned.

**Sources:** https://code.google.com/p/java-intelligent-tutor/source/browse/trunk/itjava/src/itjava/model/TFIDF.java?r=353

1. **Indexing a file:**

In Indexing I have created a singleton class so that only one object is created.

So for indexing the data structures used are: HashMap, ArrayList, and List.

To reduce the complexity I have used the concept of nested Hashmaps for retrieval of term in a word on a particular line and in a particular file.

ArrayList stores the list of files and List has the collection of files.

In Indexing catch for stopwords have been taken care of where in a particular line or phrase the stopwords are ignored.

1. **Calculating Scores:**

For calculating score of a particular word in a file we have to traverse the HashMap till we get the index of the file. Once the traversing is done the final output gives the score of a word occurred in a file and the link of a file in which it has occurred. If a particular word has appeared more times than its score must be higher.

1. **Searching Algorithm:**

During searching of a word or phrase an algorithm is used where in it traverse the whole hierarchy from getting a term to file present in the directory that is indexing. Here first when the user enters the word to search it is first stored in empty SET. Then indexing takes place step by step.

Here one more thing to note is distance matrix which is being calculated between the words and the min and max distance is calculated and stored. So if the distance is less the sentence is more relevant and should come first while searching.

At the end HashMap is returned which will have the result of the search word or phrase.

1. **Display Results:**

Finally the results are displayed in view panel wherein it keeps on sorting the words which is searched and gives the output accordingly by displaying the file and the source where the word is present using TreeMap.