Text Processing – Information Retrieval System Report

By Vijeta Agrawal

Aim:

The main focus of this report is the code implementation of the Information retrieval system that we have created based on different term weighting schemes(e.g. binary, Tf, TfIdf) , its performance analysis under different configurations and results drawn from it.

Code Implementation:

In **my_retriever** class, first of all, in **_init_** function, we are creating dictionaries **(DocumentTermDict_Binary, DocumentTermDict_tf, DocumentTermDict_tfidf)** for all three weighting schemes to calculate $(d_i)^2$ using inverted index. They store the size of the document vectors for different schemes after iterating through index file. These dictionaries are later used in **(BinaryTermWeighting, Tf_TermWeighting, Tfidf_TermWeighting)** respectively, to calculate document query similarity. We also calculate Inverse document frequency (idf) by dividing length of document list by document frequency and then taking its log.

Now in createCandidateList function, we are creating the **CandidateList** which is the union of all the documents that contain at least one of the query terms.

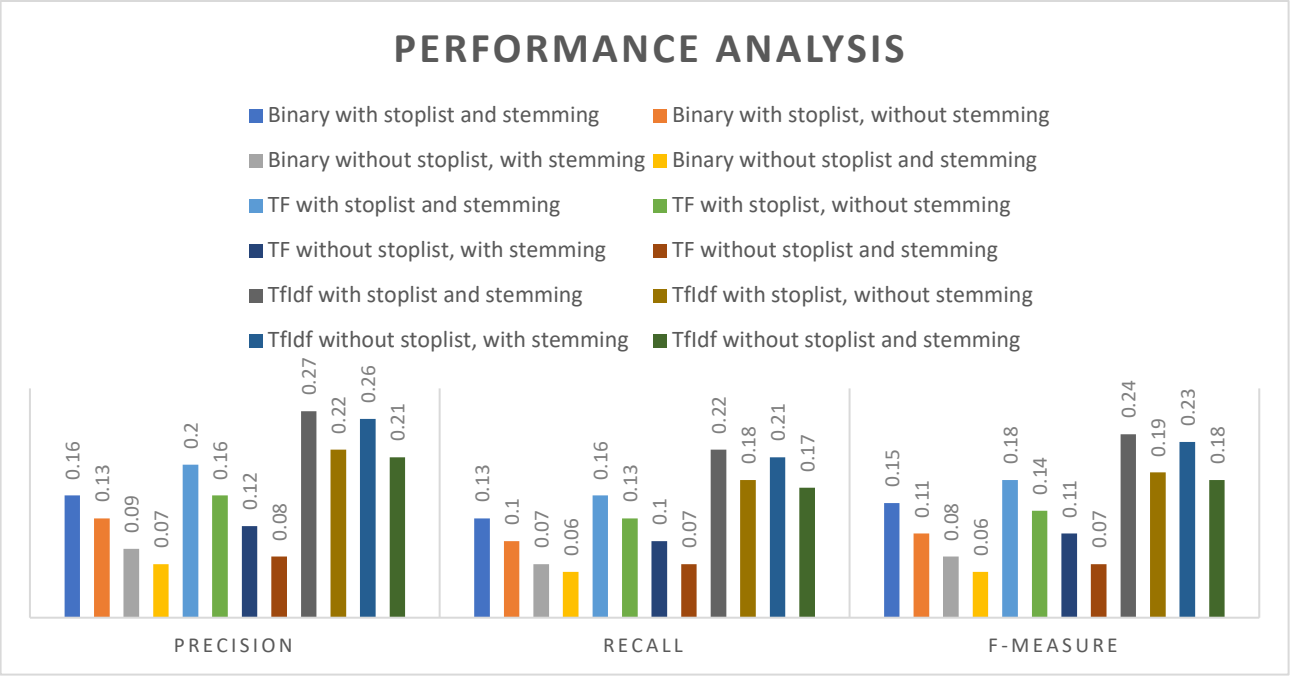
Then we create three different functions**(BinaryTermWeighting, Tf_TermWeighting, Tfidf_TermWeighting)** for document retrieval based on the weighting schemes(**Binary, Tf, TfIdf**) and pass **CandidateList** and **query** as input, these functions compute ***document query similarity*** based on ***vector space model*** according to the respective weighting schemes. The dictionaries created **(BinaryDict, Tf_Dict, and Tfidf_Dict)** are then sorted and returned, to rank the scores and get top 10 documents.

In **forQuery** function, we are performing retrieval for specified **query**, we are calling the functions we created **(BinaryTermWeighting, Tf_TermWeighting, Tfidf_TermWeighting)** with **CandidateList** and **query** as input for respective weighting schemes**(Binary, Tf, TfIdf)**.

System Performance:

To analyse the performance of the system we have used three variables (Precision, Recall, F-Measure). To observe these values better, take a look at the table and graph below:

IR System Configuration	*Precision*	*Recall*	*F-Measure*
Binary with stoplist and stemming	0.16	0.13	0.15
Binary with stoplist, without stemming	0.13	0.1	0.11
Binary without stoplist, with stemming	0.09	0.07	0.08
Binary without stoplist and stemming	0.07	0.06	0.06
TF with stoplist and stemming	0.2	0.16	0.18
TF with stoplist, without stemming	0.16	0.13	0.14
TF without stoplist, with stemming	0.12	0.1	0.11
TF without stoplist and stemming	0.08	0.07	0.07
TfIdf with stoplist and stemming	0.27	0.22	0.24
TfIdf with stoplist, without stemming	0.22	0.18	0.19
TfIdf without stoplist, with stemming	0.26	0.21	0.23
TfIdf without stoplist and stemming	0.21	0.17	0.18



Observations based on Performance Analysis:

- We can observe that our three variants (Precision, Recall, F-measure) are lowest for ***Binary without stoplist and stemming*** (0.07, 0.06, 0.06), this is because stopwords and stemming are important term manipulation practices to retrieve relevant documents. The same is the case with 'Tf and TfIdf' term weighting configurations.
- **Recall** is proportion of relevant documents returned. If we observe the recall of various configurations, we can see, it is highest for ***TfIdf with stoplist and stemming*** i.e. 0.22.
- **Precision** is proportion of retrieved documents that are relevant. If we observe the precision of various configurations, we can see, it is highest for ***TfIdf with stoplist and stemming*** i.e. 0.27.
- **F-measure** is the harmonic mean of Recall and Precision, so it combines them with equal weightage. If we observe the F-measure of various configurations, we can see, it is highest for ***TfIdf with stoplist and stemming*** i.e. 0.24.
- The configurations in which we **use Stemming and Stopwords**, produce better results than other configurations.
- The configurations using **TfIdf** term weighting, produce highest values for our variants (Precision, Recall, F-measure).

Inference:

- According to these observations, the **TfIdf** term weighting **withStemming and withStopwords** is the best configuration for relevant information retrieval because it has highest values of Precision, Recall and F-measure.
- Though the values of Precision, Recall and F-measure for **TfIdf** are highest, but still considerably lower than the ideal values of these variants i.e. 1. This implies that there is a lot of scope for improvement in this Information Retrieval System.