Registration number: 180128228

Text Processing Document Retrieval Report

Introduction

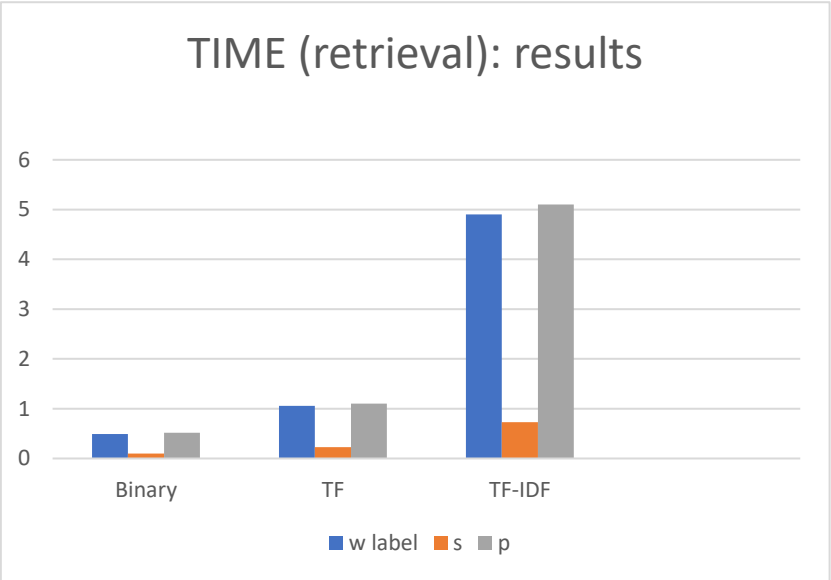
The purpose of this report is to highlight the implementation to a brief extent that I have achieved for the purpose of this assignment and I will demonstrate the term weighing schemes and how quickly the results are returned. The purpose of this assignment was to implement a basic document retrieval system and to implement different weighting schemes such as binary, term frequency (TF) and term frequency-inverse data frequency (TF-IDF).

Implementation stages

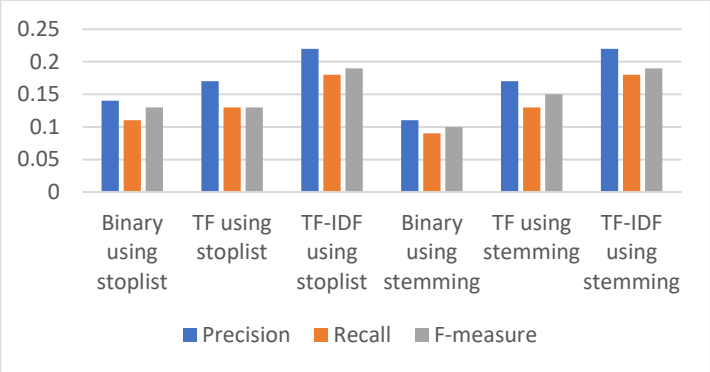
1. In the my_retriever there is a CalculateDocVector. This implementation is where it creates a new dictionary where it includes all the documents and frequencies. Furthermore, it returns the doc_vectors with is assigned to dict ().
2. I have also computed binary, this is a method for Information retrieval which makes molds the guesstimate of the document/query in terms of binary vectors. The terms presented in the document are autonomously circulated in the set of the documents furthermore, terms are also autonomously circulated in the set of the irrelevant documents. Moreover, if a term appears in a document it should be equal to 1 whereas if the term does not appear in the document it should return 0. I will be highlighting the time (retrieval) later on in this report.
3. I have implemented in my_retriever TF where as seen in the code the calculation of the query can be dropped, but if it had to be calculated it would have been calculated in following way: vQ += v*v but this is a general calculation which can be applied to all. Term frequency refers to the concept of concerning with the Information Retrieval and it highlights the results on how frequently a term/word shows in the document. Furthermore, shows the importance of the specific term/word inside the document. As seen in the code in my_retriever it also highlights that I have also calculated the document vector.
4. Regarding the TF-IDF this is an arithmetical measurement that shows how important a word is to its associated document in a given collection. It is mostly seen as a weighing factor regarding searches of Information retrieval. As seen in the code calculating the query can be dropped, however, if it was to be calculated it would be calculated in the following way: vQ += (v * math.log(totalDocs/len(doc_values)))**2.
5. I have created a set of relevant documents where it gets all the values that have at least one word relevant to our query.

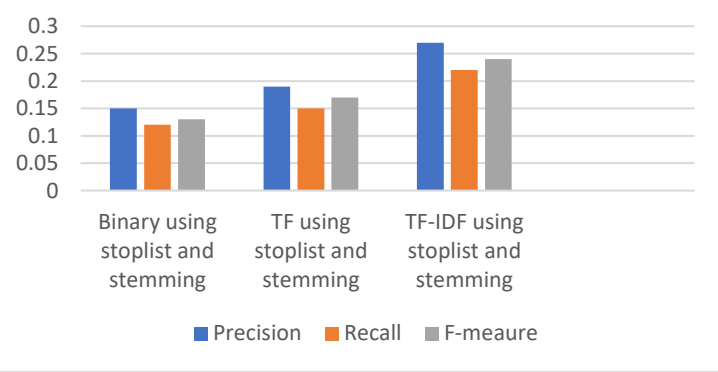
Registration number: 180128228

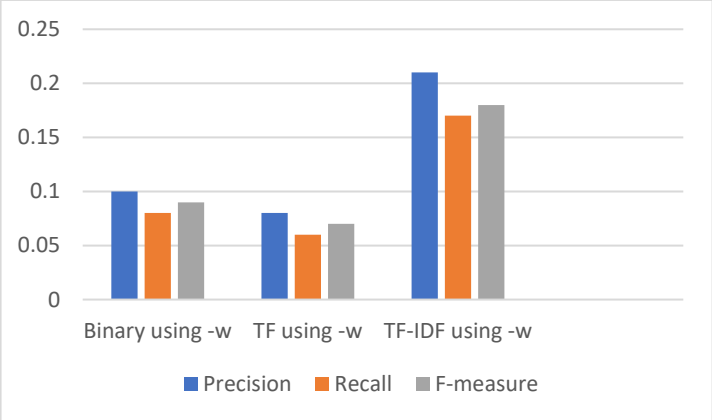
Results

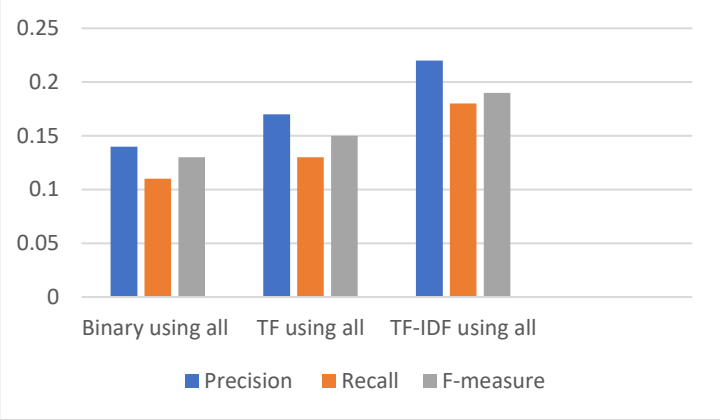


As you can see on the left-hand side the results for the TIME (retrieval) using the weighing scheme label, using the stoplist configuration and using with stemming configuration. As you can identify that using the stoplist configuration the TIME (retrieval) time is much quicker compared to the rest of the configuration. The highest number of TIME (retrieval) is the TF-IDF.









To conclude F measure could be considered as the best one this is for the reason being that F measure gets the precision and the recall into one figure as well as given equal weight to both of them.