COM3110 Assignment 1 Report – IR

1.Assignment Purpose and Aim

This assignment requires building and completing the reserved python class Retrieve from a text processing perspective by completing three models named Binary, Term Frequency (TF), and Term Frequency Inverse Document Frequency (TFIDF). For the Other classes, such as CommandLine, IndexLoader, etc., already provide data structure conversion and manipulation of files that store resultant input and output.

2. Different Schemes Analysis

2.1 Binary

The implementation is done by computing the set intersection between the set of terms in each document and the set of terms in the query, assigning a value of 1 to each element found in the intersection. Thus, if the term exists only in the document and not in the query, the term will have a weight of 0. Traversing through each document and doing so will calculate the cosine similarity score for each document. The final step is to calculate the vector size constructed for each document, thus dividing the document's score by its vector size in order to standardize the received score.

2.2 TF

For terms that are present in both the document and the query, the weight assigned to the term is (frequency of the term in the document * frequency of the term in the query). In this way, terms that appear more frequently in the document will be considered more important than other terms that appear less frequently.

2.3 TF*IDF

$$idf_{w,D} = log\frac{|D|}{df_w}$$

Figure.1

TFIDF term weighting adds an inverse document frequency element (IDF) compared to TF term weighting. This IDF is calculated by counting the inverse of each term (figure.1). This will ensure that the fact that rare terms are more important than common terms is taken into account, as it will adjust the term's score accordingly.

3.Result

F-measure	Binary	TF	TFIDF
Raw	0.06	0.07	0.18
Stemming only	0.07	0.1	0.23
Stoplist only	0.11	0.15	0.19
Both	0.14	0.17	0.24

Table 1: F-measures of IR system under a range of configurations

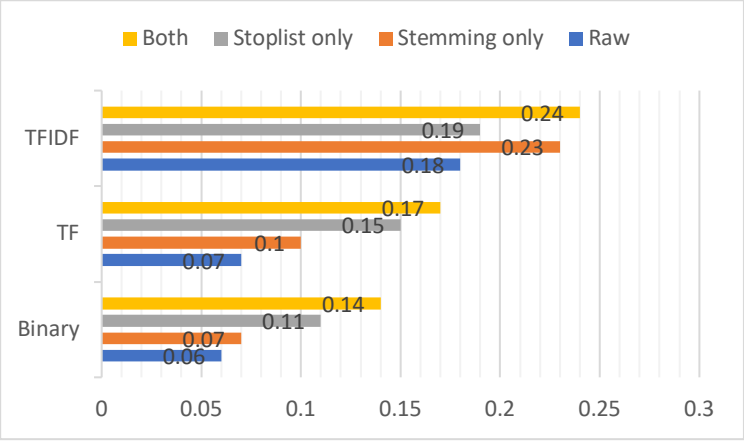

Figure 2: F-measures of IR system under a range of configurations

In this assignment, three different document retrieval term weighting algorithms were implemented, and Figure 2 and table 1 showed the F- measures scores obtained for each of the 12 configurations.

From these results we can conclude that Binary term weighting is the worst but the fastest of the three methods, while TFIDF is the best but the slowest. This is to be expected since the "binary" term weighting is the simplest method, which simply assigns 0 or 1 to each term depending on whether it is present in the query and the document. TFIDF will not only assign a binary weight to each term, it will also assign the value of how often the term appears in the document, which means that the term that appears more frequently in the document will have a final cosine similarity score for that document The impact is large. Finally, it was concluded that TFIDF term weighting is best because it does not just look at the number of times a particular term appears in a document, it also takes into account the importance of the term in the collection of documents being queried. This adjusts the weighting of terms that appear less frequently in the collection. For example, words that appear multiple times in a document but are not important are ("the", "is", etc.) These words are clearly not as useful as other words in identifying relevant documents. Multiplying the TF by the IDF score for each term takes this importance factor into account and adjusts the weighting of the terms accordingly.