**COM6115 Text Processing**

**Assignment Report**

**Student Registration Number: 190186373**

**Implementation**

In my code I was able to achieve full functionality for retrieval, returning the top 10 relevant documents to a query, judged by the system. I believe I could have utilised the Retrieve class more to better organise my code, but I still implemented the core ideas of the vector space model for information retrieval. The run time of each configuration is under 5 seconds, which seems reasonable for this task. The retrieval system was created using nested dictionaries, with just the ‘math’ library imported.

**Results**

For each configuration of retrieval using the 12 combinations of indexes and term weightings, results were produced to some level of precision. Figure 1 shows how the F-measure of my retrieval configuration differed between term weightings and pre-process methods.

|  |  |  |  |
| --- | --- | --- | --- |
| **F-MEASURE** | **Term weighting** | | |
| **Pre-process method** | **Binary** | **Term frequency** | **TFIDF** |
| **None** | 0.07 | 0.07 | 0.15 |
| **Stoplist** | 0.12 | 0.15 | 0.17 |
| **Stemming** | 0.08 | 0.1 | 0.21 |
| **Stoplist and stemming** | 0.14 | 0.17 | 0.23 |

Figure 1

We can see from Figure 1 that TFIDF was the most successful term weighting for retrieval. On average, using an index that was pre-processed with stemming and/or stoplist improved retrieval. The best retrieval performance was found when TFIDF was applied to an index with both stemming and a stoplist. With this method, an F-measure of 0.23 was achieved, just 0.1 below the example results file provided.

Furthermore, Figure 1 suggests that stoplisting causes a significant improvement to the binary and term frequency weightings, given the increase in F-measure between no pre-processing to just a stoplist pre-process. This is expected due to the removal of irrelevant words such as ‘the’ and ‘and’, which would have had a negative effect on the accuracy of binary/term frequency in terms of returning relevant documents. Since TFIDF weighting involves an inverse document frequency term, the high frequency of words such as ‘the’ and ‘and’ is accounted for already.

Figure 2: Precision (left), recall (right)

We can see similar patterns in Figure 2 as we did in Figure 1, since the F-measure is a function of precision and recall. Figure 2 shows that, generally, the retrieval system that has been implemented favours precision over recall. That is, our system has a higher proportion of retrieved documents that are relevant as oppose to the proportion of relevant documents that are turned.

Further analysis can be made by modifying the amount of documents retrieved. In this implementation, only the top 10 were returned. We can expect to see different results if we used, say, the top 5 or the top 50 most relevant documents as judged by the retrieval system.