Document Retrieval Report

Matthew Hughes

University of Sheffield

1 Description of System

The document retrieval system has been implemented as a vector space model with a variety of options to choose from. The user can choose to use a provided stoplist or a stoplist of their own choosing as well as opting to not use any stoplist. The user can decide whether or not stemming is applied to the inputs. There are three term weighting methods to choose from, the term weights in the vectors can be binary, can be the term frequencies or can be calculated using the TF.IDF approach. The system is capable of indexing a collection and storing it for future use to speed up the system.

2 Results

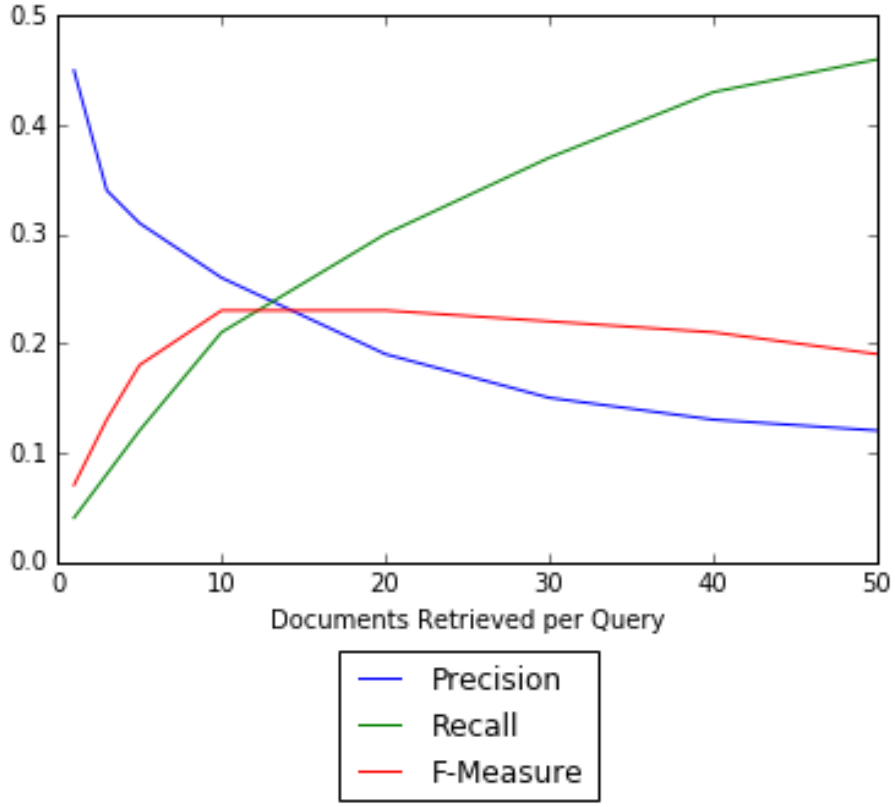
In the results table below all runs of the program retrieved 640 documents (10 per query)

Table 1: Results Table

Stoplist	Stemming	Term Weighting	Relevant Retrieved	Precision	Recall	F-Measure
False	False	Binary	47	0.07	0.06	0.07
False	False	TF	50	0.08	0.06	0.07
False	False	TF.IDF	110	0.17	0.14	0.15
False	True	Binary	62	0.10	0.08	0.09
False	True	TF	73	0.11	0.09	0.10
False	True	TF.IDF	154	0.24	0.19	0.21
True	False	Binary	83	0.13	0.10	0.12
True	False	TF	107	0.17	0.13	0.15
True	False	TF.IDF	119	0.19	0.15	0.17
True	True	Binary	99	0.15	0.12	0.14
True	True	TF	124	0.19	0.16	0.17
True	True	TF.IDF	164	0.26	0.21	0.23

From the results it can be seen that using binary term weighting consistently gives the worst performance with term frequency performing slightly better. When TF.IDF term weighing is used there is a large performance increase over the other methods. Interestingly taking the stoplist away only causes a small decrease in performance for TF.IDF but large drops in performance for TF and binary weighting. This is probably due to the words that are in the stoplist being very common and thus having a very low IDF value.

Figure 1: How changing number of documents retrieved affects Precision, Recall and F-Measure



The above results were generated with term weighting as TF.IDF and both stoplist and stemming being used. From them we can see that with low number of documents retrieved per query the the precision of the search is high but the recall is low. As the number of retrieved documents increases these measure swap places, with high recall levels but low search precision.

This makes sense as when few documents are retrieved they will all have a high level of similarity to the query which gives the high level of precision. It makes sinse that the recall rate increases as more documents are retrieved but unfortunately the precision falls as many of the documents found are not relevant to the query.

Looking at the f-measure it appears for this document collection the ideal number of documents to return per query is in the 10 to 20 range.