Report

Implementation

In implementation, it needs one extra python library which is numpy (which is for vector calculation). At first, in the constructor of class Retrieve all word terms' weight in each document would be calculated according to the option parameters. Then in the function forQuery(), the terms' weight in query would be calculated by using same methods of calculating weight in constructor. After calculating, the function calculate_similarity() would be given a matrix (constructed by the terms' of weight in relative document) and a vector(constructed by the terms' of weight in query) and then calculate the similarity between query vector and documents. Finally, the document, the top 10 of the most similar with query, would be return.

Result

Running program with the different methods of calculating terms' weight and whether use Stop List and Stemming List or not in the shell, the results have been collected and shown as below:

Binary

Stop List	Stemming List	Relevant and Retrieved	Precision	Recall	F-measure	Cost
no	no	45	0.07	0.06	0.06	12.68s
no	yes	59	0.09	0.07	0.08	13.80s
yes	no	85	0.13	0.11	0.12	3.45s
yes	yes	105	0.16	0.13	0.15	4.24s

TF

Stop List	Stemming List	Relevant and Retrieved	Precision	Recall	F-measure	Cost
no	no	49	0.08	0.06	0.07	13.33s
no	yes	72	0.11	0.09	0.10	9.69s
yes	no	106	0.17	0.13	0.15	3.30s
yes	yes	123	0.19	0.15	0.17	4.81s

TFIDF

Stop List	Stemming List	Relevant and Retrieved	Precision	Recall	F-measure	Cost
no	no	132	0.21	0.17	0.18	11.58s
no	yes	166	0.26	0.21	0.23	8.75s
yes	no	140	0.22	0.18	0.19	2.76s
yes	yes	172	0.27	0.22	0.24	3.56s

Disscussion

Three methods of the term's calculation have been implemented in program and the result is partly fit with cacm gold standard.

Comparing the results in the three different methods of calculating terms' weight, the method TFIDF is the obviously best and the second is the method TF, the worst is Binary. The reason causing this result might be that the method TFIDF is based on the method TF (which depends on the words' occurrence frequency in the document) and IDF(which is a value and negative correlation with the term's occurrence frequency, that means if term occur frequently the value would be small otherwise would be big) so that the term's weight is more reasonable. As for the method Binary, it is too simple when set the value of term's weight, so that is the worst method in three methods.

Comparing the results of using Stop List and Stemming List or not in each method, it obviously show that when Stop List and Stemming List are using together the result is the best with the highest F-measure value in each method. When it only use Stop List, the cost of running program is the smallest and the result is not good enough but usually better than using Stemming List only. When it only uses Stemming List, the result is not good enough, but only in the method TFIDF it is better than using Stop List only. If both of Stop List and Stemming List are not used, it would cost much more time and get the worst result. According to these result, Stop List and Stemming List could reduce useless terms and increase the same meaning term when doing retrieved, so that could shrinking the time of running program and improve the result.

Conclusion

In conclusion, the Stop List, Stemming List and the methods of calculating terms' weight is the crucial key to the result. In this assignment, using both Stop List, Stemming List and the method TFIDF could get the best result in all.