COM6115: Text Processing

Assignment: Document Retrieval Report

Qinzhi Zhou 210116166

Implementation

In this assignment, a Documentation Retrieval system in python was completed and been implemented to make collection files containing 10 best-ranking documents for each query in query documents.

Performance Testing and Analysis

To reduce the running time for the program, the document retrieval system applied dictionary data structure to finish the computing with reducing the for-loop operation. Which has significant influence in reducing time complexity. It can be observed that the 'tfidf' term weighting mode running time is longer because the idf term required the for-loop operation for term in documents for indexes, which has a significant influence on time complexity. It can be also observed that the stopping-list reduce more running time than stemming and perform similar in three types of weighting scheme.

Stemming can process word and its different forms are considered the to be same so it can reduce the size of words in document and the running time will be slightly reduced. Stop-list can filter the most common words which has less impact on retrieval process. According to Zipf 's Law, the common words (stop-list) is account to half of all contexts, therefore, running time results is reasonable which the Stoppling-list result is near the half in all three types of weighting scheme. The running time results is shown as figure 1.

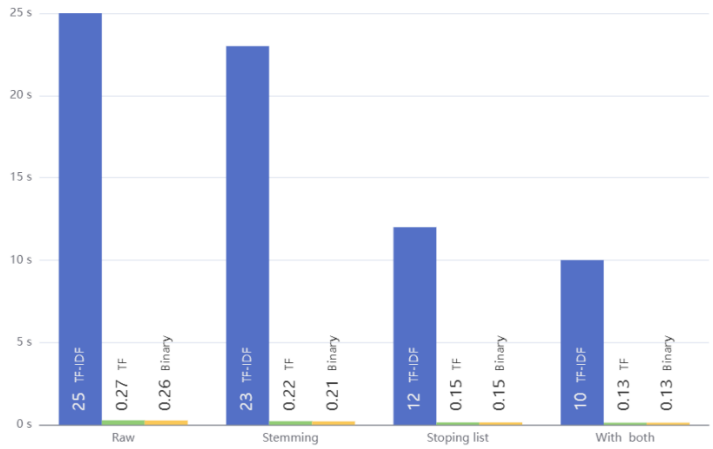

Figure 1: The running time of retrieval system

Evaluation score analysis

The evaluation score is recorded in table and visualized in figure 2 and 3. Generally the precision rank from high to low is TFIDF, TF and Binary. In Binary the precision is lowest, and the stemming seemed no effect on precision, this might be the stemming word will not change the boolean value because the existing word all present 1. The stop-list reduced the no means word so the precision

increase. In TF scheme, both stemming, and stop-list have positive impact on evaluation score because the frequencies all depend on these two methods and stop-list has greater impact because it can reduced the size of documents effectively as mentioned above as Zip's law.

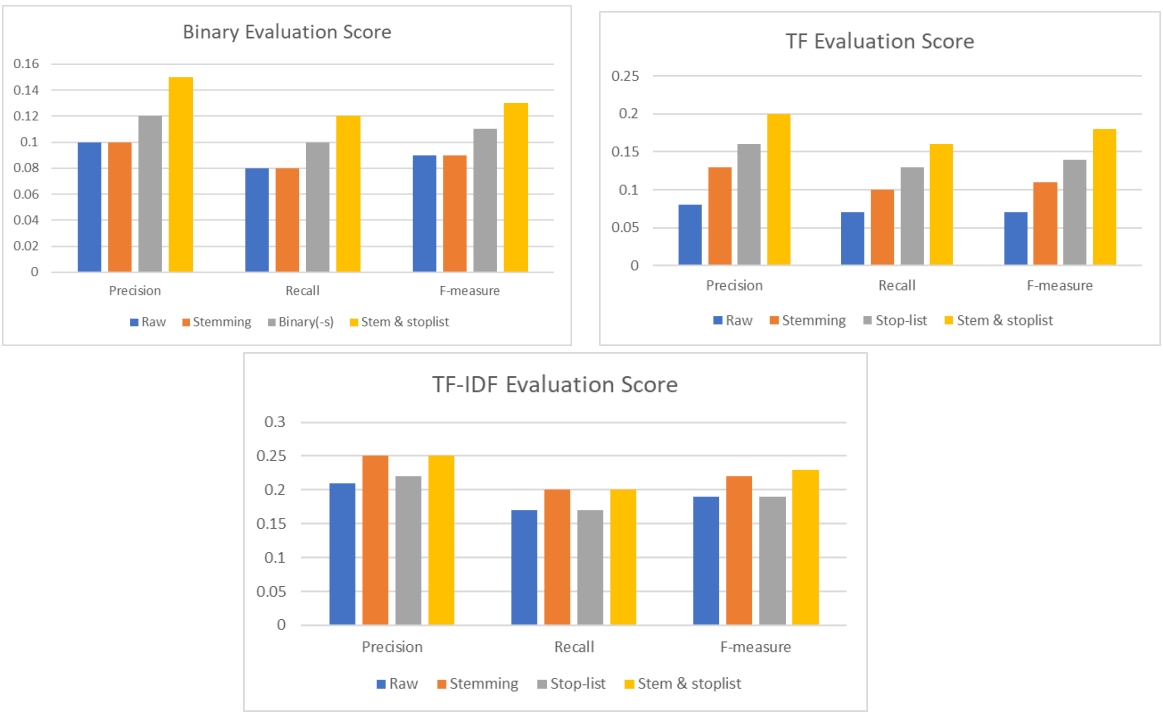

Figure 2: Evaluation scores for Binary, TF and TF-IDF

About TF-TDF, all words in document N are calculated in inverse document frequency as the idf therefore the stop-list is possible to remove some useful information, so the stop-list don't have observed positive increase on the evaluation score compared by the stemming method, so in this case, the stemming is more efficient to improve the evaluation.

Results

Weighting scheme (Mode)	*Time(s)*	*Precision*	*Recall*	*F-measure*
TF-IDF (raw)	*25*	*0.21*	*0.17*	*0.19*
TF-IDF(-p)	*23*	*0.25*	*0.2*	*0.22*
TF-IDF(-s)	*12*	*0.22*	*0.17*	*0.19*
TF-IDF (-p -s)	*10*	*0.25*	*0.2*	*0.23*
TF (raw)	*0.27*	*0.08*	*0.07*	*0.07*
TF(-p)	*0.22*	*0.13*	*0.1*	*0.11*
TF(-s)	*0.15*	*0.16*	*0.13*	*0.14*
TF (-p -s)	*0.13*	*0.2*	*0.16*	*0.18*
Binary(raw)	*0.26*	*0.1*	*0.08*	*0.09*
Binary(-p)	*0.21*	*0.1*	*0.08*	*0.09*
Binary(-s)	*0.15*	*0.12*	*0.1*	*0.11*
Binary (-p -s)	*0.13*	*0.15*	*0.12*	*0.13*

Table 1: All experimental results