

COM6115_200206297

1. Introduction

This report will show the results based on the *Retrieve* class which aims to built an IR systems. In the *Retrieve* class, three term weighting schemes are applied successful. The cosine of two vectors will be used to enlarge their similarity which will be contained in *give result* function.

2. Methods

In this part, '-s' means stoplist will be used, '-p' means stemming will be used. All results will be showed in the table below. The highest F-measures can achieve 0.28 when apply TDIDF weight, stoplist and stemming. Before giving out the results, the number of same words between query and document has been calculated and multiplied to the cosine of two vectors.

3. Performance

3.1 Binary Results

Binary weights: whether or not term is present in document.

Weighting Schemes	None	stoplist	stemming	stoplist & stemming
Rel-Retr	75	108	95	132
Precision	0.12	0.17	0.15	0.21
Recall	0.09	0.14	0.12	0.17
F-measure	0.10	0.15	0.13	0.18
Time(s)	1.166	0.355	1.269	0.503

3.2 TF Results

TF weights: number of times the word occurs in document. The calculation process will be contained in *compute TF* function.

Weighting Schemes	None	stoplist	stemming	stoplist & stemming
Rel-Retr	76	129	102	149
Precision	0.12	0.20	0.16	0.23
Recall	0.10	0.16	0.13	0.19
F-measure	0.11	0.18	0.14	0.21
Time(s)	1.730	0.487	1.767	0.724

3.3 TFIDF Results

IDF weights: inverse document frequency.

TFIDF = TF * IDF (contained in *compute_tfidf* function)

Weighting Schemes	None	stoplist	stemming	Stoplist & stemming
Rel-Retr	140	150	168	182
Precision	0.22	0.23	0.26	0.28
Recall	0.18	0.19	0.21	0.23
F-measure	0.19	0.21	0.23	0.25
Time(s)	1.729	0.602	1.880	0.796

4. Discussion

To improve the performance of models, several special methods have been applied:

- 1) To improve the precision of models, when calculating the similarity between query and document, the number of same words between query and document will be counted, and the number will be multiplied to the cosine of the query and document.
- 2) When calculating TF and TFIDF weights, we notice that for different queries, the document's TF and TFIDF weights are the same. It means we can store those weights when we first calculate them and if next time, we need to use the weights, we just need to call the answer we calculated before.
- 3) Before calculating the similarity between query and document, we could select those documents which have the same word with query as target documents. Then only calculate the similarity between query and target documents which can save lots of time.

5. Conclusion

- 1) When applying both stoplist and stemming, the system can achieve the fastest speed no matter which term weight scheme is chosen.
- 2) TFIDF can achieve better performance but it will also cost more time to give the results.
- 3) Stoplist can achieve the fastest speed and stemming will increase the time.