2/14/2021

COEN 169 Project 1

Search Systems

Vivek Ponnala

Test the performance of retrieval algorithm "RawTF" with two types of text data (i.e., raw text data and text data by stemming and removing stopwords).

1. **Evaluate the results by using "../trec_eval qrel result_rawtf" and "../trec_eval qrel result_rawtf_stemmed_nostopw". Please include the results in your report. Can you tell which result is better? If one is better than the other, please provide a short analysis. Please answer what stemmer is used in the index. Can you also use another stemmer and compare the results?**

```
[vponnala@linux10612 eval_data]$ ../trec_eval qrel result_rawtf

Queryid (Num):       30
Total number of documents over all queries
    Retrieved:     3000
    Relevant:       442
    Rel_ret:        108
Interpolated Recall - Precision Averages:
    at 0.00       0.1760
    at 0.10       0.1180
    at 0.20       0.0844
    at 0.30       0.0539
    at 0.40       0.0396
    at 0.50       0.0349
    at 0.60       0.0234
    at 0.70       0.0072
    at 0.80       0.0072
    at 0.90       0.0000
    at 1.00       0.0000
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.0449
Precision:
  At    5 docs:   0.0733
  At   10 docs:   0.0833
  At   15 docs:   0.0689
  At   20 docs:   0.0633
  At   30 docs:   0.0611
  At  100 docs:   0.0360
  At  200 docs:   0.0180
  At  500 docs:   0.0072
  At 1000 docs:   0.0036
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.0712
```

Figure 1.1: RawTF using Porter stemmer (no stemming and no removing stopwords)

```
[vponnala@linux10612 eval_data]$ ../trec_eval qrel result_rawtf_stemmed_nostopw

Queryid (Num):       30
Total number of documents over all queries
    Retrieved:     3000
    Relevant:       442
    Rel_ret:        196
Interpolated Recall - Precision Averages:
    at 0.00       0.3991
    at 0.10       0.2889
    at 0.20       0.2347
    at 0.30       0.2002
    at 0.40       0.1186
    at 0.50       0.0834
    at 0.60       0.0641
    at 0.70       0.0292
    at 0.80       0.0292
    at 0.90       0.0145
    at 1.00       0.0145
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.1174
Precision:
  At    5 docs:   0.1800
  At   10 docs:   0.1433
  At   15 docs:   0.1467
  At   20 docs:   0.1333
  At   30 docs:   0.1156
  At  100 docs:   0.0653
  At  200 docs:   0.0327
  At  500 docs:   0.0131
  At 1000 docs:   0.0065
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.1404
```

Figure 1.2: RawTF using Porter stemmer (Stemming and removing stopwords)

Based on both precision and recall values, the stemmed and no stop words query returns better results. The stemmed and no stop words query retrieves about two times the relevant documents returned for rawtf file. At all points on the interpolated recall, stemmed and no stop words attains higher average precision. However, both only retrieve less than half of the total relevant documents indicating there are many irrelevant documents. The stemmer used is the Porter stemmer. Other stemmers that could be used are Krovetz and Arabic.

```
[vponnala@linux10612 eval_data]$ ../trec_eval qrel result_rawtf_stemmed_nostopw_arabic

Queryid (Num):       30
Total number of documents over all queries
    Retrieved:     3000
    Relevant:       442
    Rel_ret:        152
Interpolated Recall - Precision Averages:
    at 0.00       0.3052
    at 0.10       0.1850
    at 0.20       0.1613
    at 0.30       0.1332
    at 0.40       0.0899
    at 0.50       0.0682
    at 0.60       0.0568
    at 0.70       0.0355
    at 0.80       0.0133
    at 0.90       0.0000
    at 1.00       0.0000
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.0859
Precision:
  At    5 docs:   0.1267
  At   10 docs:   0.1167
  At   15 docs:   0.1244
  At   20 docs:   0.1117
  At   30 docs:   0.0889
  At  100 docs:   0.0507
  At  200 docs:   0.0253
  At  500 docs:   0.0101
  At 1000 docs:   0.0051
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.1260
```

Figure 1.3: Removing stopwords and stemming for Arabic stemmer

Using the Arabic stemmer, the results for stemming and removing stopwords query is much worse than the results of the rawtf file for removing stopwords and stemming using the Porter stemmer. Not only is there lower average precision but less relevant documents are returned. This indicates that the Arabic stemmer may not be a suitable stemmer for text preprocessing.

2. **Evaluate the results by NOT removing the stopwords. A stopword list is contained in eval_data/stopwordlist. You need to modify the parameter file (e.g., remove <stopwords>stopwordlist</stopwords> in build_stemmed_nostopw_param) when apply BuildIndex. Please provide a short analysis on whether removing stopwords helps or not.**

```
[vponnala@linux10612 eval_data]$ ../trec_eval qrel result_rawtf_stemmed

Queryid (Num):       30
Total number of documents over all queries
    Retrieved:     3000
    Relevant:       442
    Rel_ret:        199
Interpolated Recall - Precision Averages:
    at 0.00       0.3959
    at 0.10       0.2933
    at 0.20       0.2386
    at 0.30       0.2045
    at 0.40       0.1194
    at 0.50       0.0834
    at 0.60       0.0641
    at 0.70       0.0292
    at 0.80       0.0292
    at 0.90       0.0145
    at 1.00       0.0145
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.1183
Precision:
  At    5 docs:   0.1867
  At   10 docs:   0.1467
  At   15 docs:   0.1489
  At   20 docs:   0.1300
  At   30 docs:   0.1144
  At  100 docs:   0.0663
  At  200 docs:   0.0332
  At  500 docs:   0.0133
  At 1000 docs:   0.0066
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.1404
```

Figure 1.4: RawTF using Porter stemmer (stemming and no removing stopwords)

When RawTF is used and stopwords are not removed, more relevant documents are retrieved. Including stopwords improves the recall and increases the precision for the documents. The interpolated average precision values for only stemming are higher compared to stemming and stopwords removed for RawTF. The average precision value is higher due to the recall precision averages at each level. Hence for RawTF in the case of stemming, removing stopwords does not help or make that much of a difference. However, without stemming, removing stopwords does make a significant difference since it increases the average precision for the documents.

Implement three different retrieval algorithms and evaluate their performance.

Preprocessing	Remove stopwords and stemming	Remove stopwords and no stemming	No removing stop words and stemming	No removing stop words and no stemming
RawTF	0.1174	0.0859	0.1183	0.0449
RawTFIDF	0.2137	0.1260	0.2137	0.1861
LogTFIDF	0.3186	0.1624	0.3179	0.2750
Okapi	0.3584	0.1727	0.3522	0.3004

Figure 1.5: Average precision values of different retrieval algorithms with different preprocessing metrics using Porter stemmer

Please compare the results and provide a short discussion about the advantage/disadvantage of the algorithms.

RawTF

For RawTF, not removing stopwords and stemming returns the highest average precision value. No preprocessing returns the lowest average precision value and removing stop words returns a slightly higher average precision value, while only stemming is comparable to stemming and removing stopwords. Calculating RawTF is quite simple. However, RawTF does not account for inverse document frequency. Overall, RawTF is the worst retrieval algorithm since it has the lowest average precision values and does not account for the number of documents or the document length and other important metrics.

RawTFIDF

For RawTFIDF, including stopwords, stemming and removing stopwords, stemming return the same highest average precision value. This is followed by including stopwords, no stemming and then removing stopwords and no stemming. Clearly, removing or including stopwords while stemming has no effect on average precision. Removing stopwords has the lowest average precision value because RawTFIDF accounts for inverse document frequency. For stemming, RawTFIDF performs well and has minimal calculation cost. Without any preprocessing done, RawTFIDF performs significantly better than RawTF. However, RawTFIDF does not account for the word semantics in a text. In comparison to RawTF, RawTFIDF has higher average precision results but still performs worse than LogTFIDF and Okapi.

LogTFIDF

For LogTFIDF, stemming and removing stopwords returns the highest average precision value. LogTFIDF performs better than RawTFIDF and RawTF in terms of precision values. At small retrieval sizes, LogTFIDF has higher recall and precision. Again, removing stopwords and no stemming has the lowest average precision value due to inverse document frequency. LogTFIDF requires stemming and stop word removal to be considered more effective and requires higher computation. However, from an implementation perspective, LogTFIDF is beneficial for small retrieval sizes due to the logarithmic complexity. LogTFIDF gives better results compared to RawTF and RawTFIDF but is still worse than Okapi.

Okapi

For Okapi, stemming and removing stop words returns the highest average precision value. Again, removing stop words and stemming has the lowest average precision value due to inverse document frequency. Okapi benefits the most from stemming and stop word removal. Okapi has the best numbers for recall and precision and shows great improvement in retrieval performance for different preprocessing techniques. Compared to other retrieval algorithms, Okapi is the best choice. However, Okapi requires extensive calculation and has overhead for information about the document's length, average document length and term frequency, etc.

Conclusion

Overall, the Okapi retrieval algorithm performs the best in comparison to RawTF, RawTFIDF, and LogTFIDF. Though Okapi requires extensive overhead, Okapi has the highest average precision values. In all retrieval algorithms except RawTF, stemming and removing stopwords produced the highest average precision values while removing stopwords and no stemming produced the lowest average precision values. This is because of inverse document frequency. In the case of RawTF, pure stemming produced higher average precision values than stemming and removing stopwords since including stop words increased recall and precision. In order to create an efficient search system, overhead for variables used should be less. Retrieval algorithms should aim to have a time complexity of constant time and try for high precision and recall. Furthermore, retrieval algorithms should aim to maximize similarity between query and document by accounting for position in text, semantics and co-occurrences of words in different documents. Perhaps, retrieval algorithms along with co-occurrence thesaurus implemented can lead to query expansion and improve search results getting higher precision and recall on average.