Matthew Koken

COEN 169

Project 1

1. 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?

|  |  |
| --- | --- |
| result\_rawtf | result\_rawtf\_stemmed |
| [mkoken@linux60806 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 | [mkoken@linux60806 eval\_data]$ ../trec\_eval qrel result\_rawtfidf\_stemmed  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 203  Interpolated Recall - Precision Averages:  at 0.00 0.6297  at 0.10 0.5090  at 0.20 0.4175  at 0.30 0.3088  at 0.40 0.2430  at 0.50 0.1906  at 0.60 0.1561  at 0.70 0.0967  at 0.80 0.0888  at 0.90 0.0421  at 1.00 0.0421  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.2286  Precision:  At 5 docs: 0.2867  At 10 docs: 0.2200  At 15 docs: 0.1822  At 20 docs: 0.1633  At 30 docs: 0.1378  At 100 docs: 0.0677  At 200 docs: 0.0338  At 500 docs: 0.0135  At 1000 docs: 0.0068  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.2368 |

Based on both precision and recall, the stemmed query returns better results. The stemmed query returned the same number of documents, however it returned almost double the number of relevant documents. At all points of interpolated recall, it achieves higher precision as well as a higher average precision over all queries. Overall, the results are much better and return more of the relevant documents that the user would want. However, both only retrieve half or less of the total relevant documents and return much greater numbers of irrelevant documents. The stemmer used is the Porter stemmer. Other stemmers available for use are the Krovetz and Arabic stemmers.

|  |
| --- |
| result\_rawtf\_stemmed\_krovertz |
| [mkoken@linux60806 eval\_data]$ ../trec\_eval qrel result\_rawtf\_stemmed\_krovetz  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 168  Interpolated Recall - Precision Averages:  at 0.00 0.3847  at 0.10 0.3030  at 0.20 0.2795  at 0.30 0.2575  at 0.40 0.1857  at 0.50 0.1629  at 0.60 0.1301  at 0.70 0.0784  at 0.80 0.0449  at 0.90 0.0242  at 1.00 0.0242  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.1616  Precision:  At 5 docs: 0.2000  At 10 docs: 0.1567  At 15 docs: 0.1289  At 20 docs: 0.1250  At 30 docs: 0.1133  At 100 docs: 0.0560  At 200 docs: 0.0280  At 500 docs: 0.0112  At 1000 docs: 0.0056  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.1833 |

Using the Krovetz stemmer, we see a decrease in the number of relevant retrieved words in comparison to the Porter stemmer. As a result, recall and precision are also lower. In this case, the Porter stemmer was more effective, though a different Krovertz dictionary could provide different results.

2. Evaluate the results by removing the stopwords.

|  |  |
| --- | --- |
| result\_rawtf\_nostopw | result\_rawtf\_stemmed\_nostopw |
| [mkoken@linux60806 eval\_data]$ ../trec\_eval qrel result\_rawtf\_nostopw  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 193  Interpolated Recall - Precision Averages:  at 0.00 0.4982  at 0.10 0.3773  at 0.20 0.2842  at 0.30 0.2057  at 0.40 0.1428  at 0.50 0.1234  at 0.60 0.0883  at 0.70 0.0474  at 0.80 0.0391  at 0.90 0.0153  at 1.00 0.0153  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.1495  Precision:  At 5 docs: 0.2067  At 10 docs: 0.1700  At 15 docs: 0.1444  At 20 docs: 0.1283  At 30 docs: 0.1189  At 100 docs: 0.0643  At 200 docs: 0.0322  At 500 docs: 0.0129  At 1000 docs: 0.0064  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.1732 | [mkoken@linux60806 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: 203  Interpolated Recall - Precision Averages:  at 0.00 0.6303  at 0.10 0.5103  at 0.20 0.4177  at 0.30 0.3105  at 0.40 0.2430  at 0.50 0.1906  at 0.60 0.1561  at 0.70 0.0967  at 0.80 0.0888  at 0.90 0.0421  at 1.00 0.0421  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.2290  Precision:  At 5 docs: 0.2867  At 10 docs: 0.2200  At 15 docs: 0.1844  At 20 docs: 0.1650  At 30 docs: 0.1378  At 100 docs: 0.0677  At 200 docs: 0.0338  At 500 docs: 0.0135  At 1000 docs: 0.0068  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.2387 |

Without stemming, removing stopwords brought a noticeable increase in the number of relevant retrieved documents. This increased both accuracy and precision across all sample points for interpolated recall and precision. However, the stemmed version did not experience the same results. The number of relevant returned documents remained the same, though there was some minor variation in recall and precision. In this case, different documents may have been chosen at different points, but the end results are nearly identical, with the no stopwords version having very small improvements. In this case, the removal of stopwords does not really have any benefit. Though the removal of stopwords for nonstemmed queries and documents provides a clear benefit.

2. Implement three different retrieval algorithms and evaluate their performance.

RawTFIDF

|  |  |
| --- | --- |
| result\_rawTFIDF | result\_rawTFIDF\_stemmed |
| [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_rawtfidf  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 197  Interpolated Recall - Precision Averages:  at 0.00 0.6303  at 0.10 0.4414  at 0.20 0.3503  at 0.30 0.2689  at 0.40 0.1734  at 0.50 0.1462  at 0.60 0.1153  at 0.70 0.0661  at 0.80 0.0476  at 0.90 0.0085  at 1.00 0.0085  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.1861  Precision:  At 5 docs: 0.2600  At 10 docs: 0.2033  At 15 docs: 0.1711  At 20 docs: 0.1583  At 30 docs: 0.1267  At 100 docs: 0.0657  At 200 docs: 0.0328  At 500 docs: 0.0131  At 1000 docs: 0.0066  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.2147 | [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_rawtfidf\_stemmed  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 245  Interpolated Recall - Precision Averages:  at 0.00 0.6552  at 0.10 0.4501  at 0.20 0.3815  at 0.30 0.2978  at 0.40 0.2438  at 0.50 0.1965  at 0.60 0.1535  at 0.70 0.1016  at 0.80 0.0825  at 0.90 0.0355  at 1.00 0.0319  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.2137  Precision:  At 5 docs: 0.3067  At 10 docs: 0.2667  At 15 docs: 0.2289  At 20 docs: 0.2100  At 30 docs: 0.1733  At 100 docs: 0.0817  At 200 docs: 0.0408  At 500 docs: 0.0163  At 1000 docs: 0.0082  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.2451 |
| result\_rawTFIDF\_nostopwords | result\_rawTFIDF\_stemmed\_nostopwords |
| [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_rawtfidf\_nostopwords  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 221  Interpolated Recall - Precision Averages:  at 0.00 0.6571  at 0.10 0.4869  at 0.20 0.3902  at 0.30 0.2955  at 0.40 0.2159  at 0.50 0.1865  at 0.60 0.1491  at 0.70 0.0972  at 0.80 0.0723  at 0.90 0.0310  at 1.00 0.0310  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.2170  Precision:  At 5 docs: 0.2867  At 10 docs: 0.2200  At 15 docs: 0.1867  At 20 docs: 0.1750  At 30 docs: 0.1444  At 100 docs: 0.0737  At 200 docs: 0.0368  At 500 docs: 0.0147  At 1000 docs: 0.0074  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.2361 | [mkoken@linux60806 eval\_data]$ ../trec\_eval qrel result\_rawtfidf\_stemmed\_nostopwords  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 261  Interpolated Recall - Precision Averages:  at 0.00 0.7009  at 0.10 0.4930  at 0.20 0.4253  at 0.30 0.3629  at 0.40 0.3141  at 0.50 0.2598  at 0.60 0.2031  at 0.70 0.1174  at 0.80 0.0854  at 0.90 0.0388  at 1.00 0.0340  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.2528  Precision:  At 5 docs: 0.3600  At 10 docs: 0.2800  At 15 docs: 0.2422  At 20 docs: 0.2200  At 30 docs: 0.1811  At 100 docs: 0.0870  At 200 docs: 0.0435  At 500 docs: 0.0174  At 1000 docs: 0.0087  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.3034 |

With a simple raw query, the RawTFIDF performs significantly better than RawTF – approaching 50% of all relevant documents. As the query and documents are refined through stemming and removal of stopwords, the performance increases dramatically. Removing stopwords brings the relevant retrieved documents up to just over half of the total number. Stemming brings further benefits, and removal of stopwords and stemming provides slightly more benefit. If the work can be put in to stem or remove stopwords from the query and documents, then RawTFIDF performs well and only carries minimal calculation cost with it. Without any processing done to the query and corpus, RawTFIDF still carries a good benefit over the base performance of RawTF.

LogTFIDF

|  |  |
| --- | --- |
| result\_logtfidf | result\_logtfidf\_stemmed |
| [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_logtfidf  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 204  Interpolated Recall - Precision Averages:  at 0.00 0.7013  at 0.10 0.5992  at 0.20 0.4292  at 0.30 0.3910  at 0.40 0.3229  at 0.50 0.2635  at 0.60 0.2159  at 0.70 0.1074  at 0.80 0.0706  at 0.90 0.0294  at 1.00 0.0294  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.2750  Precision:  At 5 docs: 0.3800  At 10 docs: 0.2667  At 15 docs: 0.2156  At 20 docs: 0.1883  At 30 docs: 0.1589  At 100 docs: 0.0680  At 200 docs: 0.0340  At 500 docs: 0.0136  At 1000 docs: 0.0068  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.2904 | [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_logtfidf\_stemmed  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 253  Interpolated Recall - Precision Averages:  at 0.00 0.8001  at 0.10 0.6441  at 0.20 0.5237  at 0.30 0.4347  at 0.40 0.3932  at 0.50 0.3175  at 0.60 0.2406  at 0.70 0.1493  at 0.80 0.1228  at 0.90 0.0646  at 1.00 0.0598  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.3179  Precision:  At 5 docs: 0.4467  At 10 docs: 0.3200  At 15 docs: 0.2978  At 20 docs: 0.2483  At 30 docs: 0.1989  At 100 docs: 0.0843  At 200 docs: 0.0422  At 500 docs: 0.0169  At 1000 docs: 0.0084  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.3660 |
| result\_logtfidf\_nostopwords | result\_logtfidf\_stemmed\_nostopwords |
| [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_logtfidf\_nostopwords  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 231  Interpolated Recall - Precision Averages:  at 0.00 0.7583  at 0.10 0.6530  at 0.20 0.5124  at 0.30 0.4419  at 0.40 0.3680  at 0.50 0.3048  at 0.60 0.2415  at 0.70 0.1561  at 0.80 0.1121  at 0.90 0.0527  at 1.00 0.0527  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.3143  Precision:  At 5 docs: 0.4000  At 10 docs: 0.3000  At 15 docs: 0.2311  At 20 docs: 0.2000  At 30 docs: 0.1733  At 100 docs: 0.0770  At 200 docs: 0.0385  At 500 docs: 0.0154  At 1000 docs: 0.0077  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.3572 | [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_logtfidf\_stemmed\_nostopwords  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 259  Interpolated Recall - Precision Averages:  at 0.00 0.7944  at 0.10 0.6739  at 0.20 0.5113  at 0.30 0.4545  at 0.40 0.3832  at 0.50 0.2999  at 0.60 0.2484  at 0.70 0.1708  at 0.80 0.1449  at 0.90 0.0659  at 1.00 0.0612  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.3245  Precision:  At 5 docs: 0.4400  At 10 docs: 0.3233  At 15 docs: 0.2778  At 20 docs: 0.2567  At 30 docs: 0.2011  At 100 docs: 0.0863  At 200 docs: 0.0432  At 500 docs: 0.0173  At 1000 docs: 0.0086  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.3644 |

With unmodified queries and documents, LogTFIDF performs slightly better than RawTFIDF and significantly better than RawTF. As with RawTFIDF, LogTFIDF receives the same benefits from stemming and stopword removal. At low sample intervals, LogTFIDF has higher recall and precision than RawTFIDF. LogTFIDF experiences the same drawbacks as RawTFIDF in terms of requiring stemming and stopword removal to be considerably more effective. It also has slightly higher computation use in the form of an additional log call, but this would be minimal. At small retrieval sizes LogTFIDF performs noticeably better – having higher recall and precision. In a case where only the first few returned documents are going to be the most seen results, then this algorithm would be a better choice than RawTFIDF.

Okapi

|  |  |
| --- | --- |
| result\_okapi | result\_okapi\_stemmed |
| [mkoken@linux60817 eval\_data]$ ../trec\_eval qrel result\_okapi  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 247  Interpolated Recall - Precision Averages:  at 0.00 0.7113  at 0.10 0.6160  at 0.20 0.5008  at 0.30 0.4279  at 0.40 0.3383  at 0.50 0.2928  at 0.60 0.2232  at 0.70 0.1590  at 0.80 0.1227  at 0.90 0.0726  at 1.00 0.0719  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.3004  Precision:  At 5 docs: 0.4267  At 10 docs: 0.3167  At 15 docs: 0.2644  At 20 docs: 0.2300  At 30 docs: 0.1833  At 100 docs: 0.0823  At 200 docs: 0.0412  At 500 docs: 0.0165  At 1000 docs: 0.0082  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.3136 | [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_okapi\_stemmed  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 291  Interpolated Recall - Precision Averages:  at 0.00 0.7977  at 0.10 0.7388  at 0.20 0.6423  at 0.30 0.5407  at 0.40 0.4033  at 0.50 0.3404  at 0.60 0.2525  at 0.70 0.1800  at 0.80 0.1246  at 0.90 0.0485  at 1.00 0.0428  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.3522  Precision:  At 5 docs: 0.4933  At 10 docs: 0.3967  At 15 docs: 0.3089  At 20 docs: 0.2900  At 30 docs: 0.2233  At 100 docs: 0.0970  At 200 docs: 0.0485  At 500 docs: 0.0194  At 1000 docs: 0.0097  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.3641 |
| result\_okapi\_nostopwords | result\_okapi\_stemmed\_nostopwords |
| [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_okapi\_nostopwords  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 255  Interpolated Recall - Precision Averages:  at 0.00 0.7035  at 0.10 0.6269  at 0.20 0.5306  at 0.30 0.4494  at 0.40 0.3673  at 0.50 0.3066  at 0.60 0.2306  at 0.70 0.1661  at 0.80 0.1221  at 0.90 0.0818  at 1.00 0.0782  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.3126  Precision:  At 5 docs: 0.4400  At 10 docs: 0.3367  At 15 docs: 0.2756  At 20 docs: 0.2383  At 30 docs: 0.1911  At 100 docs: 0.0850  At 200 docs: 0.0425  At 500 docs: 0.0170  At 1000 docs: 0.0085  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.3314 | [mkoken@linux60807 eval\_data]$ ../trec\_eval qrel result\_okapi\_stemmed\_nostopwords  Queryid (Num): 30  Total number of documents over all queries  Retrieved: 3000  Relevant: 442  Rel\_ret: 289  Interpolated Recall - Precision Averages:  at 0.00 0.7980  at 0.10 0.7130  at 0.20 0.6069  at 0.30 0.5380  at 0.40 0.4130  at 0.50 0.3458  at 0.60 0.2519  at 0.70 0.1679  at 0.80 0.1202  at 0.90 0.0497  at 1.00 0.0447  Average precision (non-interpolated) for all rel docs(averaged over queries)  0.3501  Precision:  At 5 docs: 0.4733  At 10 docs: 0.3733  At 15 docs: 0.3089  At 20 docs: 0.2850  At 30 docs: 0.2289  At 100 docs: 0.0963  At 200 docs: 0.0482  At 500 docs: 0.0193  At 1000 docs: 0.0096  R-Precision (precision after R (= num\_rel for a query) docs retrieved):  Exact: 0.3744 |

At the base level – an unprocessed query and set of documents, Okapi performs incredibly well - by far the best of any of the others. Like all of the others, it benefits from stemming and stopword removal, showing significant greatest improvements in retrieval performance. At small intervals, and even at large intervals of retrieval, Okapi shows the best numbers for recall and precision. If any processing is expected to be done to the query and documents, Okapi is the best choice despite requiring the most additional calculations. If no modification is to be done, Okapi is also the best choice. However, Okapi also requires the most overhead in information about the documents – needing to account for document length, average document length, term frequency, etc.

Overall, quite a big difference in performances can be seen. At the base level, RawTF performs the worst with RawTFIDF and LogTFIDF performing similarly. Okapi performs the best on all levels. Being the simplest to implement and least demanding in use, RawTFIDF or LogTFIDF would perform just fine for an unprocessed query and corpus. Adding only stopword removal, only stemming, or a combination of both, Okapi still performs the best across all elements. Okapi has some drawbacks in higher overhead for extra data on the documents, so if that isn't available then RawTFIDF or LogTFIDF are more useable. Recall and precision all show the best under Okapi at all intervals of document retrieval.