MSCI 541: HW 4

Jonathan Chen
20722167
November 23rd, 2021

General

All programs were created using Java (version 17, 2021-09-14 LTS, Java(TM) SE Runtime Environment (build 17+35-LTS-2724), using the IntelliJ IDE.

The topics file is named *queries.txt* and is available at the root of the repository.

To run the programs, the pre-built jar files can be used.

- To run the IndexEngine, go to the *out/artifacts/IndexEngine_jar/* directory in the command line and then run *java -jar IndexEngine.jar [path_to_latimes.gz] [output_path] [use of porter stemmer ('True'/'False')]*.
- To run the GetDoc, go to the *out/artifacts/GetDoc_jar/* directory in the command line and then run *java -jar GetDoc.jar [path to indexed data] ['id' or 'docno'] [id or docno value]*.
- To run the BooleanAND program, go to the *out/artifacts/BooleanAND_jar/* directory in the command line and then run *java -jar BooleanAND.jar [path to indexed data] [path to queries file] [output filename]*.
- To run the Evaluate program, go to the *out/artifacts/Evaluate_jar/* directory in the command line and then run *java -jar Evaluate.jar [path to indexed data] [path to qrels file] [path to results file] [output directory] [output csv filename]*.
- To run the BM25 program, go to the *out/artifacts/BM25_jar/* directory in the command line and then run *java -jar BM25.jar [path to indexed data] [path to queries file] [use of porter stemmer ('True'/'False')] [output directory]*.

Question 1

It is possible for two documents to have the same BM25 retrieval score even though one is relevant and the other is non-relevant if a more important term exists within the query and the non-relevant document contains this more important term more often than the relevant document does. Specifically, terms within the query that are in fewer documents have a higher IDF weight and are thus more influential towards the BM25 score. Therefore, if the non-relevant document has a high term frequency for this more important term, it can still produce a high BM25 score even if it does not contain other terms from the query. As a result, a relevant document that has all the terms from the query in it but has a low term frequency, can have the same score as a document that has only one term from the query in it but has a high term frequency and IDF weight for this term.

Question 2

a) The $\log(f_{ik})$ is used instead of f_{ik} so that frequent terms in a document do not completely outweigh less frequent terms, and thus creates normalization by allowing for greater equalization between terms. For example, words such as "the" and "is" naturally appear more frequently in a document but are less important than some specific words that may only appear a few times. Thus, there are diminishing returns of more frequent words where the log helps to significantly reduce the weighting of very frequent words, while only slightly reducing the weighting of infrequent words.

b) BM25 achieves a similar effect for tf in its formulation by normalizing the term in the document by the document's length. Within the BM25 equation, the term $\frac{(K_1+1)f_i}{K+f_i}$, where $K = K_1((1 - b) + b \cdot \frac{dl}{avgdl})$ represents the normalization of the term frequency. The numerator of the equation contains the term frequency, while the denominator shows the normalization against a document's length, with the variables b and K1 also being used to control the amount of normalization and saturation for the term frequency.

Question 3

Stemming speeds up retrieval speed because there are fewer words indexed. For example, the words "connect", "connects", "connected", and "connections" would all be stemmed to the word "connect," and only the word "connect" would be added to the inverted index (instead of all four words). As a result, there would be fewer terms that exist in the inverted index, so there would be fewer terms to iterate across and check for.

Question 4

Words in queries that are not found in the document collection are typically ignored. Thus, those words would have a score of 0, but the other terms in the query that are found within the document collection will still be considered.

Question 5

$\#\ of\ Documents\ =\ 131,896$
$Size\ of\ Vocabulary\ =\ 246,970$
$Matrix\ Size\ =\ 131,896\ \times\ 246,970\ =\ 32,574,355,120\ cells$
$Memory\ =\ 32,574,355,120\ \times\ 4\ =\ 130,297,420,480\ bytes\ =\ 121.35\ GB$

Therefore, if each cell of the matrix used 4 bytes, 121.35 GB of memory would be used.

Inverted Index:

$Total\ sum\ of\ all\ postings\ lists\ in\ inverted\ index\ =\ 63,779,892$

Since a postings list contains one entry for docid and another entry for count, need to divide the sum by 2:

$Total\ number\ of\ cells\ in\ matrix\ with\ values\ =\ 63,779,892\ /\ 2\ =\ 31,889,946$

$\#\ of\ Empty\ Cells\ =\ 32,574,355,120\ -\ 31,889,946\ =\ 32,542,465,174$

$Total\ Memory\ Savings\ =\ 32,542,465,174\ \times 4\ =\ 130,169,860,696\ bytes\ =\ 121.23\ GB$

Therefore, 121.23 GB would be saved if an inverted index is used.

Question 6

The aspect of Gary's design that has contributed the most to Gary's poor retrieval performance is his failure to apply the same tokenization procedures to both the queries and the documents. Since Gary removed common stopwords and applied the Porter stemmer to the queries, but not the documents, there is a mismatch between the query tokens and the document tokens. This can lead to differences in the vocabularies between the queries and the documents, and thus, certain words in the document collection cannot be found because they haven't been stemmed yet.

Question 7

d)

Run Name	Mean Average Precision	Mean P@10	Mean NDCG@10	Mean NDCG@1000	Mean TBG
baseline	0.208	0.253	0.334	0.424	1.768
stem	0.250	0.284	0.372	0.484	2.046
p-value	0.00620929	0.15523832	0.11564544	0.00244888	0.00058454

Effectiveness Measure	Best Run Score (with stemming)	Second Best Run Score (baseline)	Relative Percent Improvement	Student's t-test, two-side, paired, p-value
Mean AP	0.25	0.208	20.192%	0.00620929
Mean P@10	0.284	0.253	12.253%	0.15523832
Mean NDCG@10	0.372	0.334	11.377%	0.11564544
Mean NDCG@1000	0.484	0.424	14.151%	0.00244888

Mean TBG	2.046	1.768	15.724%	0.00058454

a. From the results above, the BM25 retrieval using the Porter stemmer had higher average scores for all 5 effectiveness measures.

b. Based on the p-values above, the difference in scores for the Mean Average Precision, the Mean NDCG@1000, and the Mean TBG are statistically significant at a 95% significance level because the p-values for those measures are less than 0.05.

c. From the figure below, green colouring indicates the topics for which the Porter stemmer had a higher score for a particular effectiveness measure (the difference is positive), while red colouring indicates the topics for which the Porter stemmer had a lower score (the difference is negative). A higher quality version of this table can be viewed in the t-test.xlsx file, available in the root of the Github directory.

Topic	AP Baseline	AP Stem	Difference	P@10 Baseline	P@10 Stem	Difference	NDCG@10 Baseline	NDCG@10 Stem	Difference		NDCG@1000	NDCG@1000	Difference	TBG Baseline	TBG Stem	Difference
401	0.0487788	0.1041077	0.0553289	0.1	0.3	0.2	0.110045883	0.246550555	0.1365047	NDCG @100	0.3515512	0.4504745	0.0989233	1.0100057	1.5947875	0.5847818
402	0.0605225	0.2061864	0.1456639	0.2	0.3	0.1	0.293455688	0.383633153	0.0901775	NDCG @100	0.2665946	0.6003825	0.3337879	1.3074061	2.5000658	1.1926598
403	0.5075374	0.5075374	0	0.6	0.6	0	0.530244205	0.530244205	0	NDCG @100	0.7407153	0.7407153	0	3.6200343	3.6200343	0
404	0.0041287	0.0098289	0.0057002	0	0	0	0	0	0	NDCG @100	0.1444771	0.1698782	0.0254011	6.11E-06	0.0079484	0.0079423
405	0.0298248	0.0259992	-0.003826	0.2	0.1	-0.1	0.142019057	0.073363922	-0.068655	NDCG @100	0.1544739	0.1469885	-0.007485	0.9959048	0.9491653	-0.04674
406	0.3919681	0.4387464	0.0467783	0.2	0.3	0.1	0.38334278	0.457491395	0.0741486	NDCG @100	0.7348802	0.7638759	0.0289957	2.1874287	2.3427376	0.1553088
407	0.2032562	0.1659461	-0.03731	0.6	0.4	-0.2	0.643403561	0.503606703	-0.139797	NDCG @100	0.574897	0.5200089	-0.054888	2.7075785	2.4589103	-0.248668
408	0.0945679	0.1356033	0.0410354	0.4	0.3	-0.1	0.538885692	0.357076477	-0.181809	NDCG @100	0.3636221	0.4616062	0.0979841	2.2027055	4.0417269	1.8390214
409	0.1	0.1	0	0.1	0.1	0	0.289064826	0.289064826	0	NDCG @100	0.2890648	0.2890648	0	0.3568845	0.3568845	0
410	1	1	0	0.4	0.4	0	1	1	0	NDCG @100	1	1	0	1.7447044	1.7447044	0
411	0.0918796	0.174399	0.0825194	0.2	0.3	0.1	0.293455688	0.444097328	0.1506416	NDCG @100	0.3672619	0.4713717	0.1041097	1.2793284	1.9732309	0.6939025
412	0.3712812	0.4685618	0.0972805	0.7	0.8	0.1	0.650033222	0.716287446	0.0662542	NDCG @100	0.6416159	0.7359097	0.0942938	6.1301129	6.5975034	0.4673905
413	0.0102041	0.0833333	0.0731293	0	0	0	0	0	0	NDCG @100	0.1508442	0.2702382	0.119394	0.0157558	0.3193749	0.3036192
414	0.0919608	0.1053803	0.0134195	0.1	0.1	0	0.167160455	0.202107347	0.0349469	NDCG @100	0.3247785	0.343575	0.0187965	0.6201176	0.5517799	-0.068338
415	0.25	0.25	0	0.1	0.1	0	0.39038005	0.39038005	0	NDCG @100	0.39038	0.39038	0	0.4696916	0.4696916	0
417	0.3360865	0.3542513	0.0181647	0.7	0.7	0	0.788549721	0.779171437	-0.009378	NDCG @100	0.7155484	0.7317348	0.0161864	3.427886	3.7777747	0.3498887
418	0.1390119	0.2600878	0.1210759	0.7	0.6	-0.1	0.785916286	0.727329844	-0.058586	NDCG @100	0.3358084	0.6177375	0.2819291	3.8617965	4.7281115	0.866315
419	0.5083991	0.575	0.0666009	0.2	0.3	0.1	0.636682439	0.74952758	0.1128451	NDCG @100	0.7352087	0.7495276	0.0143188	0.8685589	1.1776944	0.3091355
420	0.6195545	0.6171659	-0.002389	0.8	0.8	0	0.852170509	0.860381854	0.0082113	NDCG @100	0.886316	0.8863409	2.494E-05	5.0506048	4.9816572	-0.068948
421	0.0168884	0.01892	0.0020316	0	0	0	0	0	0	NDCG @100	0.2417873	0.2857011	0.0439138	0.2676954	0.2901444	0.022449
422	0.3678224	0.3780906	0.0102682	0.4	0.7	0.3	0.330874446	0.555902672	0.2250282	NDCG @100	0.6493614	0.6631105	0.0137491	7.5071364	7.8693154	0.3621789
424	0.0196692	0.1531632	0.133494	0.2	0.1	-0.1	0.139618146	0.063620788	-0.075997	NDCG @100	0.1246025	0.5917914	0.4671889	1.8632693	2.7517163	0.8884469
425	0.2844109	0.481399	0.1969881	0.6	0.7	0.1	0.631624366	0.762416492	0.1307921	NDCG @100	0.5655003	0.823749	0.2582487	4.2602578	5.168357	0.9080993
426	0.0277933	0.0342158	0.0064225	0.3	0.1	-0.2	0.222127828	0.078398269	-0.14373	NDCG @100	0.1773358	0.1767816	-0.000554	2.2589971	2.603306	0.3443089
427	0.0798961	0.0972402	0.0173441	0.2	0.2	0	0.305234884	0.305234884	0	NDCG @100	0.3041842	0.3836374	0.0794532	1.2811117	1.6187951	0.3376834
428	0.253391	0.1069568	-0.146434	0.1	0.1	0	0.39038005	0.195190025	-0.19519	NDCG @100	0.4823884	0.3319544	-0.150434	0.4693697	0.5951896	0.1258199
429	0.2811941	0.7986111	0.517417	0.1	0.4	0.3	0.39038005	0.92232608	0.531946	NDCG @100	0.5728888	0.9223261	0.3494373	0.7214239	1.6970283	0.9756044
430	0.4811326	0.6202564	0.1391238	0.4	0.4	0	0.627824847	0.660839795	0.0330149	NDCG @100	0.682275	0.7499201	0.0676451	1.6320617	1.9054408	0.2733791
431	0.0907808	0.3114672	0.2206864	0	0.6	0.6	0	0.608296529	0.6082965	NDCG @100	0.3303363	0.6788925	0.3485563	2.415576	3.4753719	1.059796
432	0.0035589	0.0016615	-0.001897	0	0	0	0	0	0	NDCG @100	0.1033628	0.0678469	-0.035516	0.1885292	0.0525236	-0.136006
433	0.0048932	0.0047139	-0.000179	0	0	0	0	0	0	NDCG @100	0.1094428	0.1085969	-0.000846	0.0203604	0.0169727	-0.003388
434	0.55	0.5434783	-0.006522	0.1	0.1	0	0.613147193	0.613147193	0	NDCG @100	0.7527426	0.7468772	-0.005865	0.687023	0.6651803	-0.021843
435	0.0611251	0.0383125	-0.022813	0.1	0	-0.1	0.078398269	0	-0.078398	NDCG @100	0.359337	0.275504	-0.083833	0.8962245	0.5953072	-0.300917
436	0.0362891	0.0858041	0.049515	0.3	0.7	0.4	0.245731329	0.677727751	0.4319964	NDCG @100	0.2008433	0.3095661	0.1087228	2.7483731	4.4411401	1.6927669
438	0.0942561	0.1143204	0.0200642	0.2	0.1	-0.1	0.135685445	0.085143118	-0.050542	NDCG @100	0.4312467	0.456156	0.0249093	1.2859747	1.4708942	0.1849194
439	0.0036008	0.0145817	0.0109809	0	0	0	0	0	0	NDCG @100	0.1253464	0.1678866	0.0425402	0.0086627	0.3328136	0.324151
440	0.5822984	0.5682618	-0.014037	0.6	0.5	-0.1	0.693663469	0.627409246	-0.066254	NDCG @100	0.8198337	0.8130702	-0.006764	2.806251	2.7608962	-0.045355
441	0.6079365	0.6496032	0.0416667	0.6	0.6	0	0.738056254	0.763401725	0.0253455	NDCG @100	0.7380563	0.7634017	0.0253455	2.3944541	2.4165875	0.0221335
442	0.025021	0.0230296	-0.001991	0.2	0.2	0	0.142795144	0.13305201	-0.009743	NDCG @100	0.2119418	0.1968467	-0.015095	1.6030193	1.3922444	-0.210775
443	0.1044737	0.0742496	-0.030224	0.2	0.2	0	0.212226366	0.13305201	-0.079174	NDCG @100	0.4277059	0.3477847	-0.079921	1.1568546	1.2516694	0.0948148
445	0.2444444	0.2444444	0	0.2	0.2	0	0.416181156	0.416181156	0	NDCG @100	0.4161812	0.4161812	0	0.8704246	0.8704246	0
446	0.0292187	0.0249431	-0.004276	0	0	0	0	0	0	NDCG @100	0.2064379	0.2332292	0.0267913	0.6972551	0.3795276	-0.317727
448	0.0177334	0.0092806	-0.008453	0	0	0	0	0	0	NDCG @100	0.2116198	0.1950473	-0.016572	0.4366729	0.0256636	-0.411009
449	0.0094681	0.0074301	-0.002038	0	0	0	0	0	0	NDCG @100	0.0967812	0.091195	-0.005586	0.3004893	0.2370169	-0.063472
450	0.2332229	0.2539825	0.0207596	0.3	0.4	0.1	0.242617855	0.444853029	0.2022352	NDCG @100	0.5923071	0.6613279	0.0690208	2.9124658	2.9706894	0.0582236

If stemming is considered to help a topic's performance if **all 5 measures** show an improvement, then those topics are:

Topic	Query Terms
401	foreign minorities, Germany

402	behavioral genetics
406	Parkinson's disease
411	salvaging, shipwreck, treasure
412	airport security
419	recycle, automobile tires
422	art, stolen, forged
425	counterfeiting money
429	Legionnaires' disease
431	robotic technology
436	railway accidents
450	King Hussein, peace

Contrastingly, if stemming is considered to hurt a topic's performance if **all 5 measures** show a worse score, then those topics are:

Topic	Query Terms
405	cosmic events
407	poaching, wildlife preserves
435	curbing population growth
440	child labor

In general, the nature of the topics where stemming helps are those that contain queries that have terms with stems in them that are related to the original query term. For example words from terms such as behavioral (stemmed to behavior), salvaging (stemmed to salvage), counterfeiting (stemmed to counterfeit), etc. all have stems that have the same meaning as their original term and thus allow for more documents to be found when the stem term is used. On the other hand, the opposite is true where stems with multiple meanings can hurt the performance. Words such as poaching (stemmed to poach) and curbing (stemmed to curb) can have multiple meanings (i.e. poach an egg or the curb of a street) so documents that are unrelated to the original query term may be matched. This then diminishes the quality of the search results because non-relevant documents are returned using the stemmed query terms.