Report

One Step Closer to Google

By

Mr. Khachen Hempatawee 5988047

Mr. Chanwit Panleng 5988076

Mr. Chaiyakarn Khanan 5988130

A Report Submitted in Partial Fulfillment of the Requirements for

ITCS414 Information Storage and Retrieval

Faculty of Information and Communication Technology

Mahidol University

Q1

Which search algorithm (Jaccard vs. TFIDF) is a better search algorithm for the LISA corpus, in terms of relevance and time consumption? Quantitatively justify your reason scientifically and statistically (i.e. avoid using your gut feelings).

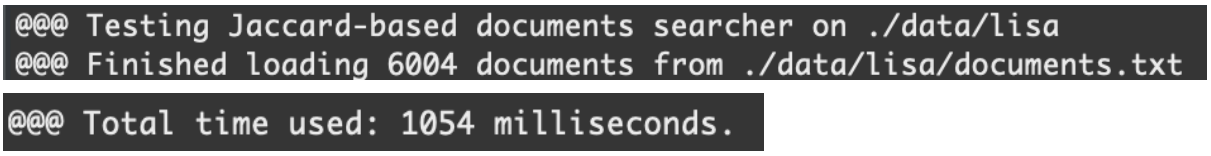
TFIDF search is better than Jaccard search. We use the result from execute the program after coding to evaluate TFIDF is better.

```
@@@ Comparing two searchers on all the queries in ./data/lisa
@@@ Finished loading 35 documents from ./data/lisa/queries.txt
@@@ Finished loading 6004 documents from ./data/lisa/documents.txt
@@@ Finished loading 6004 documents from ./data/lisa/documents.txt
@@@ Jaccard: [0.11714285714285716, 0.11296227751050206, 0.09837910465851286]
@@@ TFIDF: [0.18857142857142858, 0.2098470052155743, 0.16375909430609248]
@@@ Total time used: 22748 milliseconds.
```

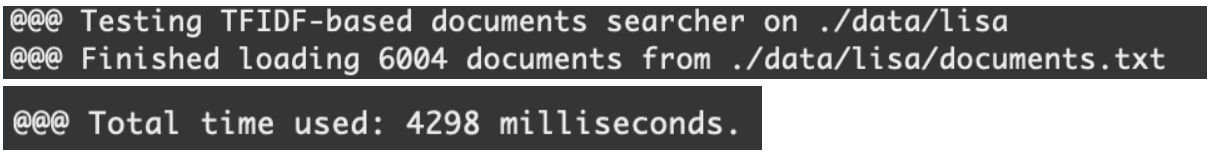
From the result above which compile all the queries, number in the basket are precision, recall and F1. We use this Three number to evaluate that TFIDF is better than Jaccard in the term of relevance because this three number can make user know percentage of correctness. TFIDF search is good to search plagiarism query token, but Jaccard search is good when the program searches the mirror query token. In conclusion of terms of relevance, TFIDF is better than Jaccard in the part of searching plagiarism or similar words. But in the part of mirror word search Jaccard is more better (This project didn’t want to find the mirror words or tokens).

Moreover, on the part of time consumption. Time consumption on the two searchers are different because of algorithms, so we show the time consumption of two searchers below.

This picture for Jaccard.



This picture for TFIDF



From the above picture, we can see that TFIDF take more time consumption than Jaccard. The reason why Jaccard compile faster than TFIDF is about calculator part. Jaccard score use solution which find only intersection word and union word between query document and each documents but TFIDF use a lot of technical calculate. TFIDF calculate solution use many methods to calculate such as document frequency, term or token frequency. All of this make TFIDF take more time than Jaccard. Moreover, TFIDF give more specific word score more than Jaccard.

Moreover, this program will be better if we combine TFIDF and Jaccard. So this type can make users or program can show the word or sentence which program want correctly. Because this will use TFIDF find the word on the first to find plagiarism word and using Jaccard to decrease the word which program don’t want. This two searcher can work well for sure.

Reference on the last paragraph

- www.sciedu.ca/air.

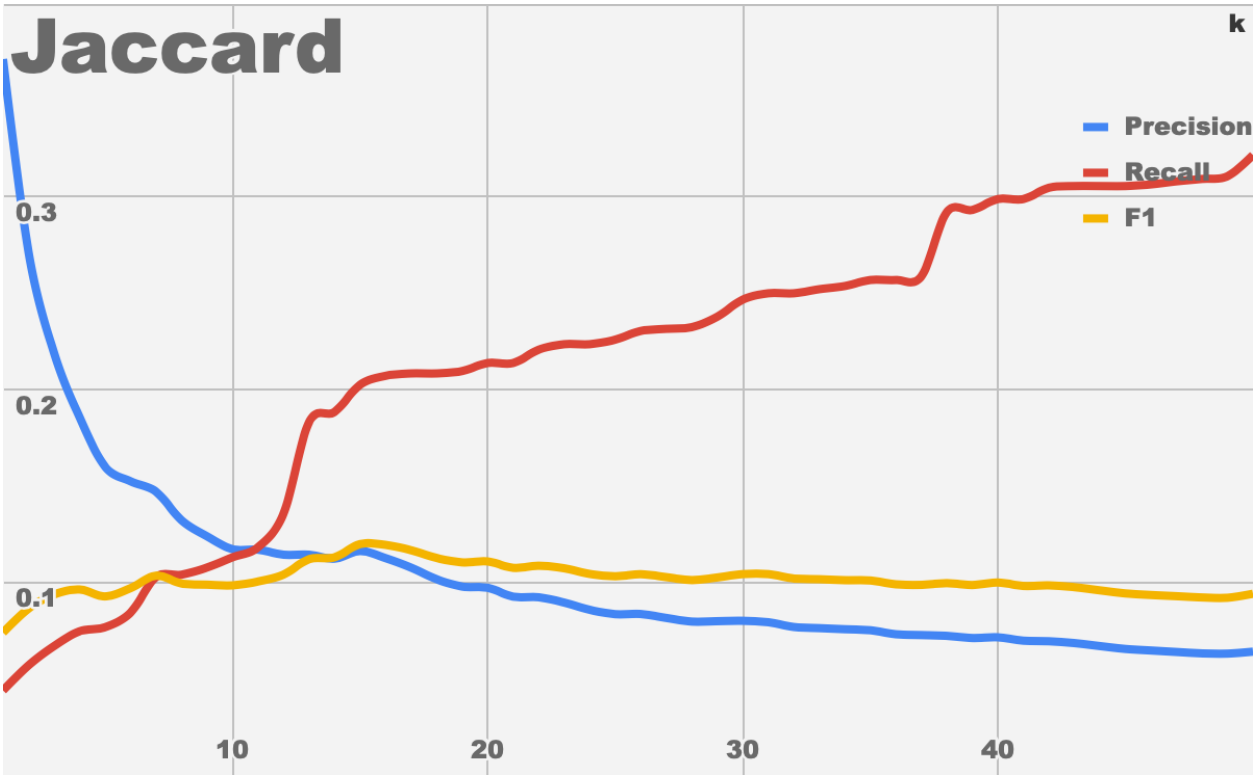
Q2

Currently, *k* is fixed at 10. Compute the average p;' decision, recall, F1 for both the search systems for each *k* (i.e. precision@k, recall@k, and F1@k), where *k* ranges from 1...50. (You should write a script that automatically does this for you, instead of manually changing k.) Visualize your findings on beautiful and illustrative plots. What conclusions can you make?

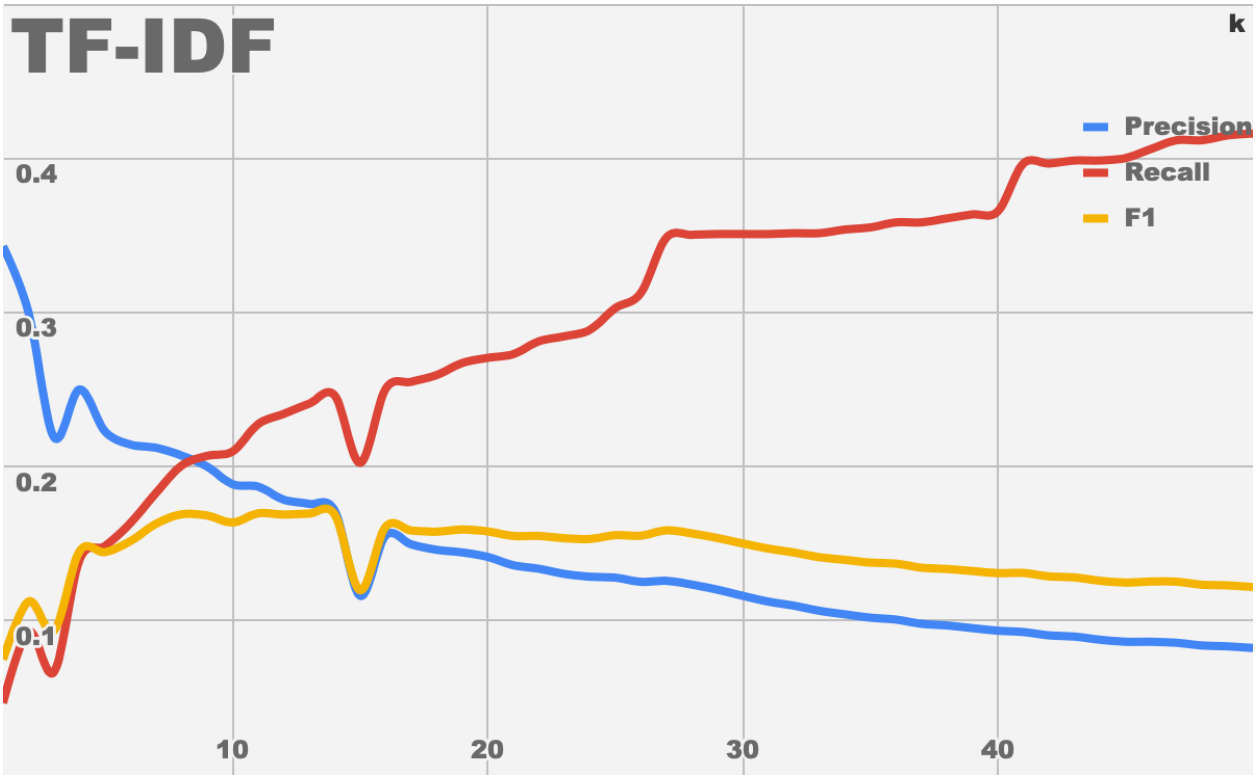
k	TF-IDF			Jaccard		
	Precision	Recall	F1	Precision	Recall	F1
1	0.3428571429	0.04666897161	0.07487020359	0.3714285714	0.04385718023	0.0738745326
2	0.3	0.09297181522	0.1125141367	0.2714285714	0.05686298961	0.08630778745
3	0.219047619	0.06686298961	0.09361631832	0.219047619	0.06686298961	0.09361631832
4	0.25	0.1394418782	0.1450059803	0.1857142857	0.07454493031	0.09625004347
5	0.2228571429	0.1484843631	0.1445498326	0.16	0.07667131545	0.09265973226
6	0.2142857143	0.1635309888	0.1517263764	0.1523809524	0.08422957032	0.09693476222
7	0.212244898	0.1829259784	0.1628991343	0.1469387755	0.1026775566	0.1034013271
8	0.2071428571	0.200573238	0.1690400196	0.1321428571	0.1039197926	0.09935068679
9	0.2	0.2069667845	0.168257058	0.1238095238	0.1076612892	0.09873887653
10	0.1885714286	0.2098470052	0.1637590943	0.1171428571	0.1129622775	0.09837910466
11	0.187012987	0.2274879219	0.169677893	0.1168831169	0.1182234506	0.1004179712
12	0.1785714286	0.2341669572	0.168866369	0.1142857143	0.1357922172	0.1040043782
13	0.1758241758	0.2407886745	0.1695243199	0.1142857143	0.1830711288	0.1119643959
14	0.1714285714	0.246198112	0.1689379322	0.112244898	0.1883949975	0.1129628343
15	0.1161904762	0.2024562593	0.1198030552	0.1161904762	0.2024562593	0.1198030552
16	0.1553571429	0.2503603839	0.160945875	0.1125	0.2070944782	0.1193826492
17	0.1495798319	0.2549811001	0.1585617451	0.1075630252	0.2083367143	0.1165760332
18	0.146031746	0.2593630536	0.1577535893	0.1015873016	0.2083367143	0.1126178035
19	0.1443609023	0.2671587362	0.1592220637	0.0977443609	0.2095789503	0.1102985066
20	0.1414285714	0.2705273573	0.1579468568	0.09714285714	0.2137461518	0.1108027295
21	0.1360544218	0.2729083097	0.1550184834	0.0925170068	0.2137461518	0.1074669312
22	0.1337662338	0.2812024953	0.1550568375	0.09220779221	0.2206344417	0.108569498

23	0.1304347826	0.2844855477	0.1535479802	0.08944099379	0.2234915845	0.1072446197
24	0.1285714286	0.2886527491	0.1530365638	0.08571428571	0.2234915845	0.1043142077
25	0.128	0.302752128	0.1555487022	0.08342857143	0.2258725369	0.1030925159
26	0.1252747253	0.3125480464	0.1551801946	0.08351648352	0.2303727985	0.104179385
27	0.1259259259	0.3483707813	0.1586453592	0.08148148148	0.2314716996	0.1026602502
28	0.1234693878	0.350270409	0.156674501	0.07959183673	0.2323375005	0.1011120395
29	0.1201970443	0.3508094926	0.1536870014	0.07980295567	0.2377644407	0.1024780341
30	0.1161904762	0.3508094926	0.1501711941	0.08	0.2466257244	0.1042279736
31	0.1124423963	0.3508094926	0.1468196648	0.07926267281	0.2499087767	0.1042234596
32	0.1098214286	0.3513485762	0.1442930535	0.07678571429	0.2499087767	0.1019502456
33	0.1064935065	0.3513485762	0.1412284764	0.07619047619	0.2520351619	0.1015626711
34	0.1042016807	0.3537295285	0.1395388773	0.0756302521	0.2536731465	0.1010642001
35	0.1020408163	0.3550900728	0.1377250563	0.07510204082	0.2568096327	0.1009138796
36	0.1007936508	0.3583731251	0.1371015012	0.07301587302	0.2568096327	0.09892209504
37	0.09806949807	0.3583731251	0.1344384978	0.07258687259	0.2587092605	0.09863035927
38	0.0969924812	0.360953025	0.1336074643	0.07218045113	0.2920425938	0.09953376776
39	0.09523809524	0.3635504276	0.1322637366	0.07106227106	0.2931414949	0.09855821834
40	0.09357142857	0.3655912439	0.1309069385	0.07142857143	0.2988104291	0.09986207445
41	0.09271777003	0.3965436249	0.1310515568	0.06968641115	0.2988104291	0.09807572774
42	0.09054421769	0.3965436249	0.1287889562	0.0693877551	0.304671235	0.09838499363
43	0.08976582125	0.3985083268	0.1281364512	0.06843853821	0.3055370359	0.09744010224
44	0.08772418058	0.3985083268	0.12596075	0.06688311688	0.3055370359	0.09579284516
45	0.08640826873	0.4000956284	0.1247669384	0.0653968254	0.3055370359	0.09420199075
46	0.086391869	0.4059564343	0.1254483654	0.06459627329	0.3064028367	0.09338795074
47	0.08576832151	0.4116253685	0.1253662732	0.06382978723	0.3079901383	0.09277143556
48	0.08398033126	0.4116253685	0.1233800567	0.0630952381	0.3092323743	0.0921157743
49	0.08343154891	0.4147618547	0.122970058	0.06297376093	0.3106372588	0.09194768756
50	0.08233333333	0.4160040907	0.1218697224	0.064	0.3216525328	0.0940331997

Jaccard



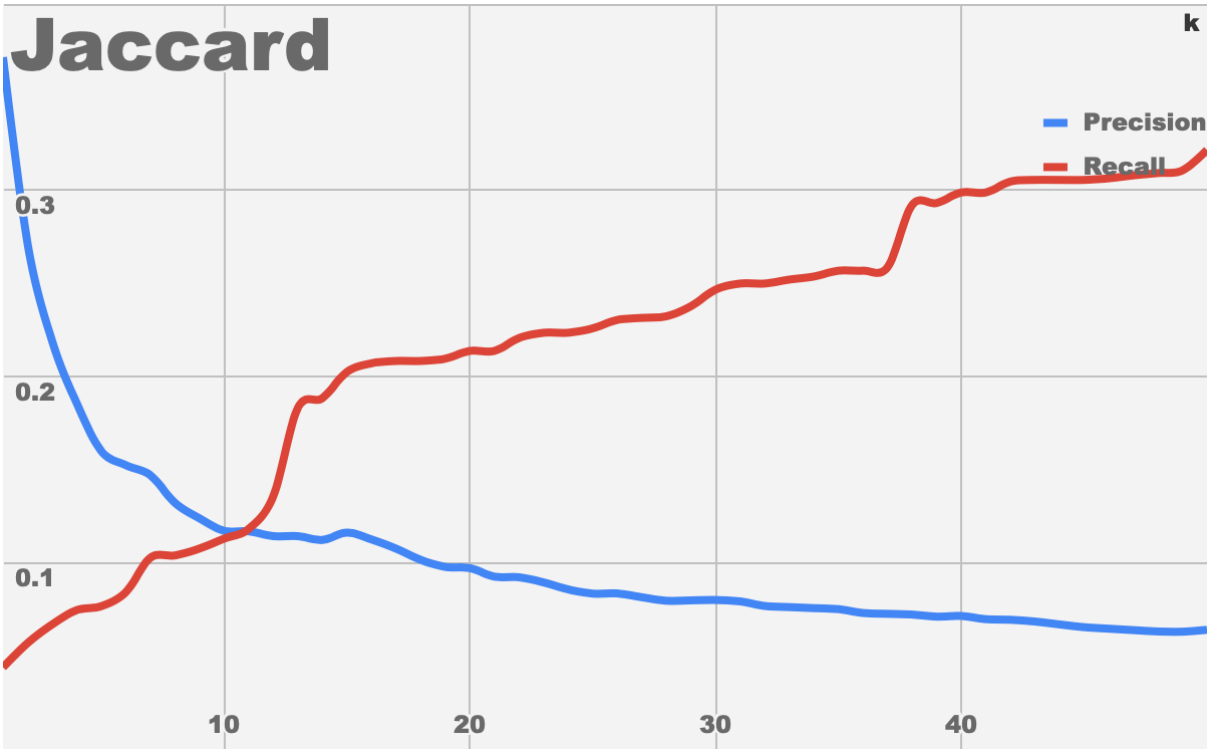
TFIDF

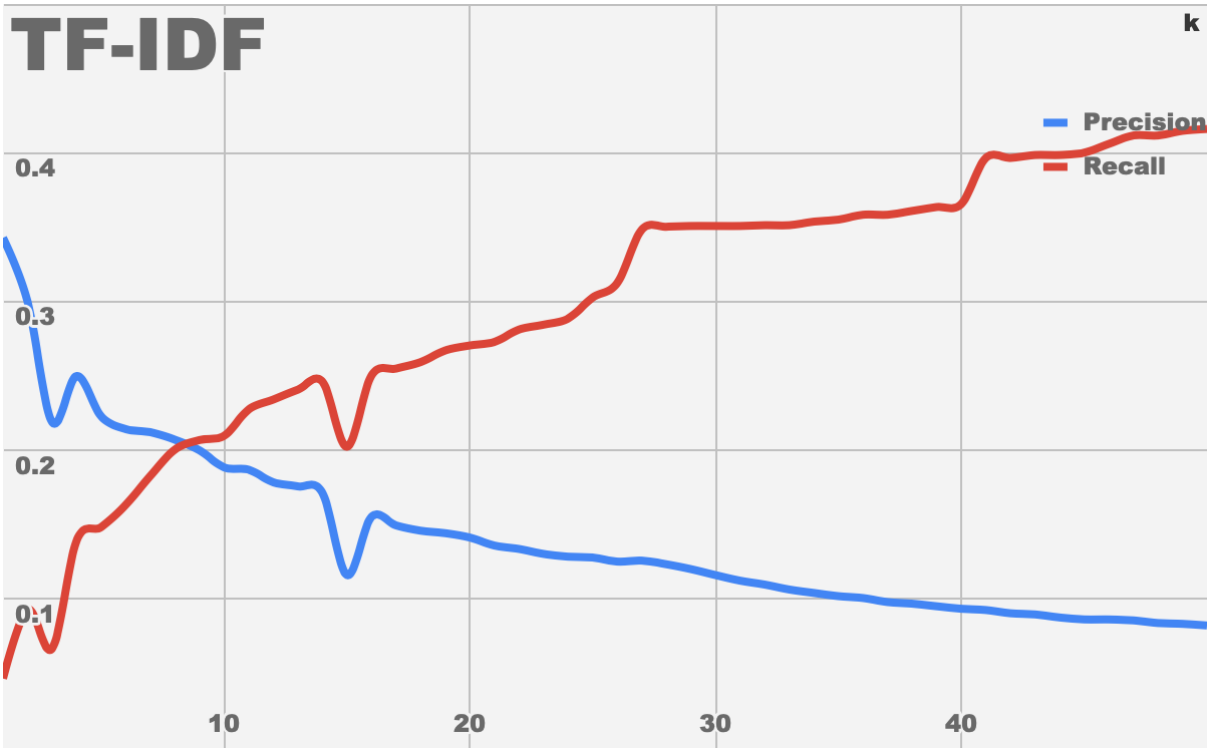


From graph and table, we can conclude that Jaccard better than TF-IDF when K values (Making the program show list of highest score k values, so this make k is the number which list how many highest score should show) is less than ten. Because Jaccard searcher is good on the part of word or token which have the same in the documents. But TFIDF is good on the plagiarism word. So we can see that on the less k values is want the same word not similar word, this make Jaccard work better. Moreover, on the highest k values word on query and documents are hard to be same, but it can similar. So this make TF-IDF work better than Jaccard. All of this compare, we use F1 values to compare this two searcher because F1 is the values which show correctness of this two searcher. F1 values means more correctness, if F1 closer to one. But if F1 closer to zero, it means this fail.

Q3

From Q2 generate precision vs recall plots for each search system. Explain how you can use these plots to explain the performance of each search algorithm.





On this question we talk about performance on this two searcher. So we can conclude this two searcher by precision which specific recall. Form graph, we can see that on the less of k value or less recall values can make this two searcher have more precision. But Jaccard have more precision than TFIDF searcher on the case which k values less than ten. Because Jaccard is good searcher on the way which want the same words or the less recall. TFIDF is better when high k values or high recall values because this case or k and recall high values want to give score or find the word which can plagiarism (similar word or same word). All of this depend on precision which means numbers of correctness result divided by number of correct when compare with the full correct result. And Recall which means numbers of number of correct divided by all relevant sample.