Yet another comparison of Lucene and Indri performance

Howard Turtle
Center for Natural Language Processing
Syracuse University
Syracuse, NY 13078
turtle@syr.edu

Yatish Hegde
Center for Natural Language Processing
Syracuse University
Syracuse, NY 13078
yhegde@syr.edu

Steven A. Rowe
Center for Natural Language Processing
Syracuse University
Syracuse, NY 13078
sarowe@syr.edu

ABSTRACT

We present results that compare the performance of Lucene and Indri at two points in time (2009 and 2012) using data from TREC 6 through 8. We compare indexing throughput, index size, query evaluation throughput, and retrieval effectiveness. We also examine the degree to which the results produced by the two systems overlap with an eye toward estimating the performance increase that might be expected by combining the results of the two systems.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Search Process; H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Open source search engines, information retrieval, performance evaluation, fusion of search results

1. INTRODUCTION

We have used a number of open source and proprietary search engines to support research at the Center for Natural Language Processing often combining the results from multiple engines to good effect [2]. The two engines we most often use are Lucene/Solr (http://lucene.apache.org/solr/) and Indri [6].

Lucene/Solr is attractive because it is a relatively full feature package that makes it easy to field Web-based applications. Indri is attractive because it offers better search results and because it offers a highly expressive query language that allows very fine grained control of a search. The engines are also natural choices because we are familiar with them. One author (Rowe) is a Lucene contributor and chair of the Lucene Project Management Committee. A second author (Turtle) has contributed to the development of Indri and Lemur.

In order to test the assertion that Indri produced better rankings, to assess the likelihood that combining Lucene and Indri results would improve overall performance, and to improve our understanding of the relative performance of the two engines we ran a series of experiments in 2009 to compare performance on TREC data. Both engines have evolved since 2009 so we reran the tests this year to evaluate any change. We present the results of both sets of experiments here.

2. PRIOR WORK

Lin [3] compared the performance of Lucene and Indri as part of a study of the impact of retrieval quality on the performance of question answering systems and concluded that there was no significant difference between the ranking quality of the two systems. Lin used relatively long queries, which may account for the performance similarity as Indri performance is known to degrade with query length. This study also uses early versions of Indri and Lucene.

Middleton and Baeza-Yates [4] conducted an early survey of the features of 17 search engines and conducted extensive performance tests with 12 of those engines, including Lucene and Indri. Their tests used TREC data (Disk 4, WT10g) and reported indexing time, index size, query evaluation time (one and two word queries), and retrieval effectiveness although not all engines participated in all of the tests. The Middleton and Baeza-Yates study used early versions of Lucene (1.9.1) and Indri (2.4); both engines have changed significantly since their study.

Perea-Ortega et al [5] compare the ranking performance of three retrieval systems (Lucene, Lemur, and Terrier) when used in a Geographical Information Retrieval (GIR) system. They used the GeoCLEF 2007 data and ran both mono- and bilingual queries. They conclude that Lemur works best for monolingual queries and that Terrier works better for bilingual queries.

Armstrong et al [1] compared the retrieval effectiveness of five search engines, including Indri and Lucene (version 2.4), using TREC data from 1994 to 2005. Queries were based on title plus description fields, similar to the long query experiments described in Section 3. They found a somewhat smaller difference between the two systems than reported here – for TREC 6 and TREC 8 data they report that Lucene's Mean Average Precision (MAP) scores are 3% to 4.5% lower than Indri whereas our experiments show MAP scores to be 5.6% lower. Differences in the Lucene version used and details of the experimental setup (e.g., stopwords, stemmer) likely account for the difference.

3. EXPERIMENTS

Two sets of experiments were run to compare Indri with Lucene/Solr performance at two points in time. The first set, originally run in October of 2009 but repeated on more modern hardware, compares the versions of Indri and Lucene that were current at the time. The second set compares the

SIGIR 2012 Workshop on Open Source Information Retrieval, August 16, 2012, Portland, Oregon, USA
ACM X-XXXXX-XX-X/XX/XX.

	TREC Disk 4	TREC Disk 5	Total
Number of documents	293,710	262,367	556,077
Collection size (Mb)	1,194	945	1,344
Number of queries	150	150	

Table 1: Collection statistics

versions of Indri and Lucene that were current in June of 2012. Out-of-the-box settings were used for both systems with no tuning or special query formulation.

While our focus is on the performance of the Indri and Lucene search engines, the experiments are run using their respective wrappers, Lemur and Solr. In 2009, the current version of Lemur was 4.10 which used Indri version 2.10. The current version of Solr was 1.4 which used Lucene version 2.9.1. By June 2012, the Lemur software had been repackaged so that the wrapper software and search engine were combined in a single distribution, Indri 5.3. In 2012, Solr and Lucene remained separate packages but the version numbers had been aligned so the current version of Solr was 3.6 which used Lucene version 3.6.

We collected performance information on indexing speed, index size, query evaluation times for two query sets, ranking performance for those queries (using trec_eval), and overlap between the results produced by the two systems. All experiments were run on an Intel(R) Xeon(R) CPU E5335 @ 2.00GHz (four cores) running Debian Squeeze v6.0.4 and Java 1.6 (Oracle). While the test system was a four core system, all tests were single threaded. The test system is equipped with 16Gb of memory but both Indri and Solr were only given 1Gb.

3.1 Data

We used a single data set consisting of TREC disks 4 and 5 for both sets of experiments. The Porter stemmer and the default Solr stop word list (35 words) was used for both the Indri and Lucene collections. Collection statistics are shown in Table 1.

3.2 Queries

Two sets of queries were generated from TREC topics 301-450 (TREC 6 through 8). The first query set (short queries) consists of the text from the title element of the TREC topics. The short queries average 2.6 words per query. The second set (long queries) consists of the text from both the title and description elements with an average length of 18.7 words per query. The queries were completely unstructured and made no use of proximity or other special query language features.

4. RESULTS

4.1 Indexing

The results of the indexing experiment are shown in Table 2. The index sizes remained the same for both systems between 2009 and 2012. Both systems produce indexes that are roughly twice the size of the source file. Indri produced a more compact index; the Solr index is roughly 17% larger than the Indri index. The indexing time results are quite different. Indri 5.3 indexing time increased from Indri 4.1 by about 10% whereas Solr indexing time decreased between the two versions by about 24%. Indri indexing is faster than Solr for both experiments but the difference is greatly reduced, Solr indexing was slower by a factor of 1.7 in 2009 but only by a factor of 1.2 in 2012.

4.2 Query evaluation

Query evaluation times are shown in Table 3. There is little difference in the query throughput of the two engines for short queries and no change in performance between 2009 and 2012 for short queries. For long queries there are significant differences. Indri is significantly slower than Lucene for long queries. The increase in time for evaluating long vs short queries is between 20 and 25 for Indri (long queries are roughly 7 times longer than short queries) and only a factor of 2 for Lucene. Indri query evaluation for long queries slowed between 2009 and 2012.

Note that for these tests the entire index for each of the systems was cached by the operating system as the test machine was equipped with 16Gb of memory and the combined size of the two indexes is only 5.2 Gb. Each system was run once to prime the OS cache then run 3 to 5 times to gather timings. Comparing performance when the the collections must be read from disk is for future work.

Indri is much more processor intensive than Lucene. During the experiments Indri used essentially 100% of a CPU core whereas Lucene used roughly 50%. Indri generally used less memory – for the 2012 versions running long queries Indri used up to 45Mb of memory whereas Lucene used 150 to 300Mb.

4.3 Retrieval effectiveness

Retrieval effectiveness results are shown in Tables 4 (2009) and 5 (2012). For short queries, Indri produces a significantly better ranking. Performance as measured by MAP is 44% less for Lucene. Using precision at fixed ranks of 10 and 20, Lucene performance is roughly 30% lower, using bpref Lucene is 26% lower. For long queries, the differences are smaller but Indri still produces noticeably better rankings – Solr is 16% lower using MAP and 14% lower using bpref.

The change in retrieval effectiveness between 2009 and 2012 is shown in Tables 6 (Indri) and 7 (Solr). For both systems the change is small. Indri showed no change for short queries and mixed results for long queries (slight increase in P10 and P20). Solr showed small improvements for short queries and mixed results for long queries.

4.4 Overlap

	Short queries		Long queries	
	Indri 5.3	Solr 3.6	Indri 5.3	Solr 3.6
P(in OL)	0.2076		0.4653	
P(+|in OL)	0.4824		0.4688	
P(-|in OL)	0.4363		0.4546	
P(?|in OL)	0.0813		0.0767	
P(+)	0.3721	0.2587	0.4167	0.3813
P(-)	0.5112	0.5967	0.5127	0.5547
P(?)	0.1167	0.1447	0.0707	0.0640
P(+|not OL)	0.3577	0.2057	0.3596	0.3040
P(-|not OL)	0.5253	0.6450	0.5697	0.6429
P(?|not OL)	0.1170	0.1493	0.0707	0.0531

Table 8: Overlap in top 10 ranks

	2009		2012	
	Lemur 4.1	Solr 1.4	Indri 5.3	Solr 3.6
Index size (gigabytes)	2.4 (1.8x)	2.8 (2.1x)	2.4	2.8
Indexing time (sec)	863	1,461	942	1,113
Throughput (Mb/sec)	1.6	0.9	1.4	1.2

Table 2: Indexing results

	2009		2012	
	Lemur 4.1	Solr 1.4	Indri 5.3	Solr 3.6
Short queries (sec)	10	13	10	13
(sec/query)	0.07	0.09	0.07	0.09
Long queries (sec)	200	25	251	25
(sec/query)	1.33	0.16	1.67	0.16

Table 3: Query evaluation times

	Short queries			Long queries		
	Lemur 4.1	Solr 1.4	Change	Lemuri 4.1	Solr 1.4	Change
MAP	0.1951	0.1092	−44.1	0.2235	0.1840	−17.7
Precision at 10	0.3713	0.2573	−30.7	0.4053	0.3827	−5.6
Precision at 20	0.3247	0.2173	−33.1	0.3500	0.3220	−8.0
bpref	0.2219	0.1645	−25.9	0.2449	0.2081	−15.0

Table 4: Indr vs. Solr retrieval effectiveness (2009)

	Short queries			Long queries		
	Indri 5.3	Solr 3.6	Change	Indri 5.3	Solr 3.6	Change
MAP	0.1948	0.1098	−43.6	0.2224	0.1856	−16.1
Precision at 10	0.3707	0.2607	−29.7	0.4167	0.3813	−8.5
Precision at 20	0.3243	0.2207	−31.9	0.3590	0.3183	−11.3
bpref	0.2219	0.1645	−25.9	0.2433	0.2087	−14.2

Table 5: Indr vs. Solr retrieval effectiveness (2012)

	Short queries			Long queries		
	Lemur 4.1	Indri 5.3	Change	Lemur 4.1	Indri 5.3	Change
MAP	0.1951	0.1948	−0.2	0.2235	0.2224	−0.5
Precision at 10	0.3713	0.3707	−0.2	0.4053	0.4167	+2.8
Precision at 20	0.3247	0.3243	−0.1	0.3500	0.3590	+2.6
bpref	0.2219	0.2219	0.0	0.2449	0.2433	−0.7

Table 6: Change in Indri retrieval effectiveness over time

	Short queries			Long queries		
	Solr 1.4	Solr 3.6	Change	Solr 1.4	Solr 3.6	Change
MAP	0.1092	0.1098	+0.5	0.1840	0.1856	+0.9
Precision at 10	0.2573	0.2607	+1.3	0.3827	0.3813	−0.4
Precision at 20	0.2173	0.2207	+1.6	0.3220	0.3183	−1.1
bpref	0.1645	0.1645	0.0	0.2081	0.2087	+0.3

Table 7: Change in Solr retrieval effectiveness over time

The overlap between the results produced by both systems is shown in Tables 8 and 9 (+ means document judged relevant, − means document judged not relevant, ? means document not judged, OL means in overlap). These numbers are important for two reasons. First, effectiveness results can be biased in favor of a system that has been used extensively in the TREC experiments if many of the documents retrieved by the other system have not been judged

	Short queries		Long queries	
	Indri 5.3	Solr 3.6	Indri 5.3	Solr 3.6
P(in OL)	0.2356		0.4973	
P(+|in OL)	0.4042		0.4026	
P(-|in OL)	0.5211		0.5325	
P(?|in OL)	0.0747		0.0649	
P(+)	0.3274	0.2207	0.3590	0.3183
P(-)	0.5562	0.5880	0.5763	0.6133
P(?)	0.1163	0.1913	0.0647	0.0683
P(+|not OL)	0.3026	0.1515	0.3093	0.2199
P(-|not OL)	0.5750	0.6334	0.6283	0.7119
P(?|not OL)	0.1224	0.2151	0.0624	0.0681

Table 9: Overlap in top 20 ranks

and will therefore be treated as not relevant. The results in Table 8 suggest that this is not a factor in these experiments – the number of unjudged documents is small for both systems with between 85% and 90% of all documents judged for short queries and between 90% and 95% for long queries.

Second, they provide an indication of how much improvement might be achieved by combining the results of the two systems. If the overlap is large then the combined result can have little increase in recall so the primary source of improvement is the reordering of the documents based on combined score. If the overlap is smaller then the probability that a randomly selected document from the overlap is relevant is an indication of how well a simple voting strategy might work. For example, in Table 8 roughly 20% of the documents retrieved by the two systems with short queries were the same. The probability that a document retrieved by both systems is relevant is 0.4824 which is significantly higher than the probability of relevance achieved by either system individually (0.3721 for Indri and 0.2587 for Solr).

5. CONCLUSIONS

The results presented here allow direct comparison of the two search engines. It also allows comparison of the changes in the two engines between 2009 and 2012.

Index size for the two engines did not change between 2009 and 2012. Indri produces a somewhat smaller index (1.8 times as large as the source collection) than Lucene (2.1 times). In terms of indexing throughput, Indri declined between between 2009 and 2012 (from 1.6Mb/sec to 1.4Mb/sec) whereas Lucene performance improved (from 0.9Mb/sec to 1.2Mb/sec). In 2012, Indri still enjoyed a slight advantage over Lucene (1.4Mb/sec vs 1.2Mb/sec).

In terms of query throughput there is little difference between the two engines for short queries but Indri is significantly slower than Lucene for long queries.

In terms of retrieval effectiveness, Indri results are significantly better than Lucene results especially for short queries. Using precision at rank 20, Lucene rankings are roughly 30% worse for short queries and roughly 10% worse for long queries. Retrieval effectiveness did not change significantly for either engine between 2009 and 2012, at least for the simple queries used in these experiments.

The overlap results show that the documents retrieved by the two engines are significantly different, especially for short queries. Using the top ten documents retrieved, for short queries roughly 80% of all documents retrieved appear in only one of the two rankings. For long queries, about half of the documents retrieved appear in only one ranking. The overlap results also show that even simple strategies for combining results can yield significant improvements in retrieval effectiveness.

6. REFERENCES

[1] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Has adhoc retrieval improved since 1994? In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 692–693, New York, NY, USA, 2009. ACM.

[2] J. W. Keeling, E. E. Allen, S. A. Rowe, A. M. Turner, J. A. Merrill, E. D. Liddy, and H. R. Turtle. Development and evaluation of a prototype search engine to meet public health needs. In *Proceedings of the American Medical Informatics Association*, pages 693–700, 2011.

[3] J. Lin. The role of information retrieval in answering complex questions. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 523–530, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[4] C. Middleton and R. Baeza-Yates. A comparison of open source search engines. Technical report, Universitat Pompeu Fabra (Barcelona, Spain), Oct. 2007. Available at: http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf.

[5] J. Perea-Ortega, M. Garcia-Cumbreras, M. Garcia-Vega, and L. Urena-Lopez. Comparing several textual information retrieval systems for the geographical information retrieval task. In E. Kapetanios, V. Sugumaran, and M. Spiliopoulou, editors, *Natural Language and Information Systems*, volume 5039 of *Lecture Notes in Computer Science*, pages 142–147. Springer Berlin / Heidelberg, 2008.

[6] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language-model based search engine for complex queries. Technical Report IR-407, CIIR, Department of Computer Science, University of Massachusetts Amherst, 2005.