CIIR Experiments for TREC Legal 2007

Howard Turtle
CogiTech
Jackson, WY
turtle@cogitech.com

Donald Metzler*
Yahoo! Research
Santa Clara, CA
metzler@yahoo-inc.com

Abstract

Four baseline experiments using standard Indri retrieval facilities and simple query formulation techniques and two experiments using more advanced formulations (dependence models and pseudo-relevance feedback) are described. All of the experiments perform substantially better than the median performance of automatic runs but exhibit lower estimated precision and recall at B than the reference Boolean run.

1 Introduction

Information retrieval techniques have been widely embraced by the legal community. Large scale commercial IR systems (Lexis and Westlaw) have been in use since the 1970's. Legal retrieval systems were among the first to adopt advanced natural language and ranking techniques – West Publishing incorporated the inference network models developed at the University of Massachusetts in the early 1990's [TC90, Gri92]. Lexis incorporated vector space techniques in the mid 1990's [PS95].

Early commercial systems focused largely on case law, statutes, and other materials that were generated as part of a print publication process. Legal discovery was slower to develop because most of the historical materials of interest were in paper files and the cost of document conversion was high. By the early 1980's, though, research was underway to evaluate the effectiveness of discovery using then extant tools [BM85, Dab86]. Today the historical materials of interest are increasingly electronic. As the 2006 Legal Track organizers point out the "importance of doing well at e-discovery is hard to overstate" [BLO06].

Recall is important for legal retrieval. Not citing an important precedent or statute can seriously weaken an attorney's argument. Not finding a needed document in discovery can seriously weaken a case. The tools necessary to evaluate high recall retrieval are not well developed or understood. The Legal Track evaluation measures represent an attempt to better quantify recall for large test collections.

2 Experiment Description

The TREC Legal collection used for the CIIR experiments was built using standard Indri [SMTC04] tools. A small program was written to extract text from the XML source and convert it to normal TREC format for input to the Indri build utility (IndriBuildIndex). The size of the text extracted from the 61.25 Gb XML source was 52.5 Gb. Using the extracted file the build of 6,910,192 documents took 22.3 hours on an Intel Xeon 3.0GHz box (four processors but the build software is single threaded). The collection was indexed with a three word stoplist (a, of, the) and the Krovetz stemmer.

We did not participate in the 2006 TREC Legal Track so the bulk of our work was to establish baseline performance numbers for future work. We submitted four runs using standard Indri retrieval facilities and basic query formulation and two runs to test advanced techniques (Table 1).

2.1 Baseline retrieval experiments

The four baseline runs submitted all use standard Indri retrieval facilities (using IndriRunQuery) and differ only in the query formulation techniques used. With the exception of UMass10 all of the runs were produced automatically from the XML topic source.

The first run (UMass11) consists of queries formed from words and phrases extracted from the RequestText topic element. Phrases were recognized using two phrase dictionaries (one con-

*Work conducted while at the Center for Intelligent Information Retrieval.

		Description
UMass10	Manual	Manually edited version of UMass11 queries.
UMass11	Automatic	Terms and phrases extracted from RequestText.
UMass12	Automatic	Queries generated from FinalBooleanQuery.
UMass13	Automatic	UMass11 and UMass12 combined.
UMass14	Automatic	Dependence model features extracted from RequestText.
UMass15	Automatic	Pseudo-relevance feedback using UMass14 for initial retrieval.

Table 1: Experimental runs

taining Wordnet colocations, the other a legal dictionary). Phrases were implemented using the Indri unordered word operator. Simple synonym expansion was done to expand hyphenated terms (e.g., high-phosphate expands to #syn(high-phosphate #2(high phosphate))) and using a small thesaurus containing common expansions (e.g., United States expands to #syn(#1(united states) america usa)). Stop structures and common stop prefixes (e.g., "Please produce all documents") were removed. This query set also includes one date filter (topic 89). Date filters (e.g., #dateafter(12/31/1980)) were useful with the training topics (four queries used them) but were not useful with the test topics.

The second run (UMass10) is a hand-edited version of the UMass11 queries. Modifications to the queries were made to add phrases that were not recognized using the phrase dictionaries, to drop noise terms that were not recognized as stop structures, and to expand the range synonym classes. No documents were reviewed during the revision process.

UMass12 consists of queries automatically generated from the FinalBoolean query contained in the XML topic using simple syntactic translations. Quoted strings were converted to phrases. OR groupings were converted to synonym classes. Proximity operators were retained. Truncation and wildcard operations were converted to the most plausible stem form. The objective here was not to replicate the Boolean query evaluation but, rather, to capture some of the value added when human searchers created the Boolean queries (phrases, synonyms, term variations).

UMass13 consists of a weighted combination of the UMass11 queries (RequestText) and the UMass12 queries (FinalBoolean). Weights were set based on training set performance.

The four baseline query sets were run on the same four processor Xeon box (3.0 GHz) used to build the collections. Query sizes and evaluation times are shown in Table 2 for the case where 25,000 documents are ranked (as required by the track) and where 25 documents are ranked (closer to interactive use). Note that the query evaluation code is multithreaded and can use all four processors effectively.

2.2 Term dependence and feedback experiments

In addition to the four baseline retrieval experiments, we performed two advanced runs that focused on term dependence and pseudo-relevance feedback. These runs are intended to improve both precision (via modeling term dependencies) and recall (via pseudo-relevance feedback).

The UMass14 run uses Metzler and Croft's dependence model, which is based on a Markov Random Field (MRF) model for information retrieval [MC05]. The model captures various types of dependencies by modeling features over sets of query terms. These dependence features include single term features, phrase features, and general term proximity feature such as ordered and unordered window matching. The model has been shown to significantly improve precision-oriented measures, especially on large data sets.

The model was applied to the RequestText in the following manner. First, a set of stop words, stop prefixes, and stop phrases were applied to the text in order to remove non-informative terms. The remaining terms were then treated as the query and the sequential dependence variant of Metzler's dependence model was applied [MC05]. Using this variant, only dependencies between adjacent query terms are modeled.

For example, for topic 63, we first distill the query *exclusivity clause sugar contract* from the RequestText. Next, dependence model features, such as `#1(exclusivity clause)` (exact phrase) and `#uw8(sugar contract)` (unordered window of size 8), are extracted and used for ranking documents.

	depth 25000		depth 25	
	CPU sec	Elapsed sec	CPU sec	Elapsed sec
UMass10	648	163	530	133
UMass11	1211	305	979	246
UMass12	1487	374	1424	358
UMass13	3084	774	2915	732

Table 2: Query evaluation performance (46 queries)

No parameter tuning was done on the MRF model. Instead, we used parameter settings that were known to be effective in the past. Previous experiments have shown that the model is relatively insensitive across collections. Thus, our parameters are likely to be reasonable.

The UMass15 run builds upon the UMass14 run by adding an additional pseudo-relevance feedback (i.e., query expansion) step to the process. Our pseudo-relevance feedback approach, called Latent Concept Expansion (LCE), is designed to complement the MRF model and has been shown to improve recall [Met07]. The technique is similar in nature to Lavrenko and Croft's relevance models [LC01], except the underlying probability distribution over query terms and documents is no longer treated as a bag of words multinomial model.

Although the details of the technique are beyond the scope of this paper, we provide a high level overview of how the approach works. First, an initial retrieval is done using the approach described for our UMass14 run. Then, a set of k concepts (such as terms, phrases, etc.) are extracted from the top N ranked documents. These concepts are then added to the original query. The augmented query is then evaluated to produce the final ranked list of results.

We use $N = 5$, $k = 5$, and only consider single term concepts for expansion. A more rigorous exploration of parameter values and possible expansion entities was not done due to time constraints.

3 Results and discussion

Experimental results are shown in Table 3. For each of the six experimental runs we show estimated recall (Est R@B) and estimated precision (Est P@B) at the cutoff determined by the reference Boolean run (B), precision at rank 5, precision at rank 20, and bpref. Precision at rank 20 is not a reliable measure (judging was only done to a depth of 5) but it is included since it gives a rough indication of what an interactive user might see on the initial result screen.

We also show the results for the reference Boolean run (refL07B) and the median value for the 25 runs that used only the RequestText element. Note, however that UMass12 and UMass13 are not directly comparable – UMass12 uses only the FinalBoolean query, and UMass13 uses both RequestText and FinalBoolean.

All of the UMass runs perform substantially better than the corresponding medians, they outperform the reference Boolean on normal precision and bpref measures, but they do not perform as well as the reference Boolean on estimated precision and recall at B.

The manually reviewed queries (UMass10) exhibit higher precision than the automatically produced version based on RequestText (UMass11) but slightly worse estimated recall at B. Of the baseline runs, the queries based on the combination of RequestText and FinalBoolean exhibited the best overall performance.

The two advanced runs (UMass14 and UMass15) using only RequestText performed significantly better than the corresponding baseline (UMass11) on all measures. They also performed better than the best baseline (UMass13) that used both the RequestText and FinalBoolean elements.

4 Conclusion

Overall, the UMass experimental runs performed well on all of the traditional IR measures. None of our runs (and, it would appear, none of the automatic runs) performed as well as the reference Boolean on estimated precision and recall at B. We don't really know how to interpret this result as yet – this marks the first use of these estimated measures.

One issue of concern with this test collection is the shallow pooling depth. It will be difficult to

	Est R@B	Est P@B	P@5	P@20	bpref
UMass10	0.1310	0.2068	0.4884	0.2779	0.3168
UMass11	0.1367	0.1911	0.3767	0.2453	0.2962
UMass12	0.1502	0.1971	0.3442	0.2093	0.2853
UMass13	0.1618	0.1945	0.4372	0.2872	0.3163
UMass14	0.1472	0.2078	0.4605	0.3198	0.3271
UMass15	0.1650	0.2286	0.4605	0.3279	0.3446
req25 median	0.1078	0.1796	0.3535	0.2314	0.2839
refL07B	0.2158	0.2921	0.0326	0.0326	0.2598

Table 3: Retrieval Effectiveness (43 queries)

compare results using traditional measures between this topic set (pooling depth of 5) and other TREC results (e.g., pooling depth of 100 for the 2006 Legal Track topics).

The Legal Track is an important step toward a better understanding of high recall retrieval but much work remains. We need to understand and validate the new measures developed for this year. We also need to figure out how these results might find their way into a set of real world discovery tools that operate with diverse electronic collections in addition to historical OCR materials.

Acknowledgments

This work was supported by the Center for Intelligent Information Retrieval and NSF grant number CCF-0205575.

References

[BLO06] Jason R. Baron, David D. Lewis, and Douglas W. Oard. Trec-2006 legal track overview. In Elen M. Voorhees and Lori P. Buckland, editors, *The Fifteenth Text Retrieval Conference Proceedings (TREC 2006)*, pages 1–2. National Institute of Standards and Technology, 2006. Proceedings available as NIST Special Publication 500-272.

[BM85] D. C. Blair and M. E. Maron. An evaluation of retrieval effectiveness for a full-text document retrieval system. *Communications of the ACM*, 28(3):290–299, 1985.

[Dab86] Daniel P. Dabney. The curse of Thamus: An analysis of full-text document retrieval. *Law Library Journal*, 78:5–40, Winter 1986.

[Gri92] Cary Griffith. WESTLAW's WIN: Not only natural, but new. *Information Today*, pages 9–11, October 1992.

[LC01] Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In *Proceedings of SIGIR 2001*, pages 120–127, 2001.

[MC05] Donald Metzler and W. Bruce Croft. A Markov random field model for term dependencies. In *Proceedings of SIGIR 2005*, pages 472–479, 2005.

[Met07] Donald Metzler. Latent concept expansion using Markov random fields. In *Proceedings of CIKM 2007*, page To appear, 2007.

[PS95] Teresa Pritchard-Schoch. Comparing natural language retrieval: WIN & FREESTYLE. *ONLINE*, 19(6):83–87, July 1995.

[SMTC04] Trevor Strohman, Donald Metzler, Howard Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*, 2004.

[TC90] Howard Turtle and W. Bruce Croft. Inference networks for document retrieval. In Jean-Luc Vidick, editor, *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 1–24. ACM, September 1990. Reprinted in *Readings in Information Retrieval*, Karen Sparck Jones and Peter Willett (eds.), 1997.