MultiText Legal Experiments at TREC 2008

Thomas R. Lynam and Gordon V. Cormack
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

Abstract

Our TREC 2008 effort used fusion IR methods identical to those used for our TREC 2007 effort; in addition we used logistic regression to attempt to learn the optimal K value for the primary $F_1@K$ measure introduced at TREC 2008. We used the Wumpus search engine combining several methods that have proven successful, including cover density ranking and Okapi BM25 ranking, and combination methods. Stepwise logistic regression was used to estimate K using TREC 2007 results as training data.

1 Introduction

For the legal track, we created several base runs using various primitive IR approaches that have worked well previously, then combined these base runs to improve performance. This work is very similar to our previous year's work for TREC legal task [LBCC04, LC06, CL07, Tom07]). The one major addition this year was the use of logistic regression to learn the optimal K value.

2 Legal Retrieval Model

Our legal retrieval effort consists of three parts:

1. creating eight base runs using multiple query fields and several information retrieval (IR) methods
2. fusing the results of the results of the base runs (or subsets of the base runs) together
3. learning K values to optimize $F1@K$ scores.

2.1 Base Runs

We created seven base runs as well as using the provided TREC Boolean run. Table 1 shows the ranking and IR methods for the base runs. Six of the runs use Okapi BM25 ranking. Three of the runs use character 4-grams instead of words as features. One run uses cover density ranking(CDR). Porter stemming is perform in one run.

Character 4-grams were used in order to mitigate the large number of errors in the legal track corpus which is made up of documents scanned from images on which optical character recognition OCR was performed. This has cause the documents to be what a photographer would describe as "noisy". There are many incorrectly recognized letters and words. N-gram retrieval was used to lessen this problem of "noisy" documents. We know from previous experience that character 4-grams are competitive with bags of words for our IR techniques, and had reason to believe that they might be more robust to the errors introduced by OCR. Furthermore, we know that character 4-grams provide much better performance for spam filtering.

Using the FinalQuery and RequestText fields seven different queries were created; one for each base run. Table 2 shows the queries produced for topic 110, whose RequestText field is:

> Please produce all reports, written memoranda, correspondence, and other documents related to employment safety standards.

Base Run	Ranking	IR method
Boolean	-	
relaxed_boolean	CDR	
okapi_requesttext	BM25	
okapi_requesttext_stem	BM25	stem
okapi_booleantext	BM25	
4-gram_okapi_requesttext	BM25	4-grams
4-gram_okapi_requestwords	BM25	4-grams
4-gram_okapi_booleantext	BM25	4-grams

Table 1: IR methods

Descriptions and rationale for each of the eight base runs are detailed below.

Base Run	Field	Query
relaxed_boolean	FinalQuery	(employ! OR job! OR occupation! OR profession! OR work! OR trade!) and ((safety) and (standard! OR criteri! OR measure! OR norm OR norms OR rule! OR requirement! OR law! OR statute! OR regulation!))
okapi_requesttext	RequestText	employment safety standards
okapi_requesttext_stem	RequestText	employment! safety! standards!
okapi_booleantext	FinalQuery	employ job occupation profession work trade safety standard critieri measure norm norms rule requirement law statute regulation
4-gram_okapi_requesttext	RequestText	Zemp empl mplo ploy loym oyme ymen ment entZ ntZs tZsa Zsaf safe afet fety etyZ tyZs yZst Zsta stan tand anda ndar dard ards rdsZ
4-gram_okapi_requestwords	RequestText	Zemp empl mplo ploy loym oyme ymen ment entZ Zsaf safe afet fety etyZ Zsta stan tand anda ndar dard ards rdsZ
4-gram_okapi_booleantext	FinalQuery	Zemp empl mplo ploy loyZ Zjob jobZ Zocc occu ccup cupa upat pati atio tion ionZ Zpro prof rofe ofes ...

Table 2: Queries for topic 110 (Z represents space)

boolean

The boolean results supplied with the TREC 2008 corpus were used for this base run, in the order provided.

relaxed_boolean

Our implementation of boolean retrieval, which ranks results by relevance and also includes (at low rank) documents that match a weakened version of the query. This run was ranked using cover density ranking (CDR), the approach that MutliText has used with success over the years for IR and QA [CCKL00, CCL01, CC96]. CDR searches for short intervals of text containing important terms from the query. The highest-level disjuncts (or conjuncts) from the boolean queries are removed. For example, the query

> ("smoke" or "cigarette") and ("girl" or "boy")

was considered to have two terms:

> ("smoke" or "cigarette") ("girl" or "boy")

The effect is that documents matching more terms or terms that are closer together are ranked before those matching fewer terms or terms that are farther apart.

okapi_requesttext

This base run used okapi BM25 [RWJ+95] document ranking on the RequestText field.

okapi_requesttext_stem

This run is the same as okapi_requesttext but a Porter stemmer was used on the text

okapi_booleantext

The FinalQuery field was converted to a bag of words by stripping out the boolean operators. Okapi BM25 document ranking was completed using the stripped bag of words.

4-gram_okapi_requesttext

The RequestText field is converted to 4-grams and treated as a bag of words. For example, the phrase

> "smoke it"

was considered to have terms

> "smok" "moke" "oke " "ke i" "k it"

The 4-gram bag of word queries are issued against the corpus using the okapi BM25 document ranking.

4-gram_okapi_requestwords

This run is very similar to 4-gram_okapi_requesttext but the 4-grams did not span over words. If we look at the previous example

> "smoke it"

was considered to have terms

> "smok" "moke" "it"

4-gram_okapi_booleantext

Similar to the okapi_booleantext run all formating and boolean operators was stripped from the FinalQuery field. The remaining text is converted to 4-grams like the 4-gram_okapi_requestwords run.

2.2 Fusion Method

We exploited the known performance improving technique of combining multiple methods (fusion) for all our submitted runs. Based on our TREC 2007 legal findings [CL07] the fusion of base runs was done using the CombMNZ[SF94, BKFS95] combination method. CombMNZ is a common method of combining multiple retrieval schemes. It combines and re-scores all documents for each query from a set of retrieval schemes. The fused document score is the sum of the scores for the given document of the schemes multiply by the number of schemes the document appeared.

2.3 Optimizing K

We experimented with a linear regression for learning optimal K values. As input we used the following fields:

number of terms in the RequestText field
number of text terms in the FinalQuery
number of brackets in the FinalQuery
number of OR in the FinalQuery
number of terms in the relaxed query
B
score@rank=1
score@rank=5
score@rank=10
score@rank=20
score@rank=50
score@rank=500
score@rank=5000
score@rank=25000
score@rank=last
average score

Using stepwise logistic regression applied to our TREC 2007 results we found that the most significant indicator fields were score@rank=5, score@rank=10, and score@rank=20, and that the other fields contributed little. For our final runs we only used these fields. We experimented with linear and logarithmic transfer functions and found neither to be consistently superior. Therefore we submitted different runs using the two methods as well as the average of the results of the two. In order to train using the TREC 2007 results, it was necessary to simulate runs containing 100000 documents. This we did by merging together the separate results of our eight runs from 2007.

In addition, we noted that high values of K yielded results that were about as good as the learned values. We therefore included runs for which K was arbitrarily fixed to 25000 (the maximum value for TREC 2007) and 100000 (the maximum value for TREC 2008).

2.4 Submitted Runs

Submitted runs are described below. Six of the runs – wat1fuse, wat4fuse, wat5fuse, wat6fuse, wat7fuse, and wat8fuse – differed only in the values chosen for K and Kh. That is, each consisted of exactly the same documents in exactly the same order. Table 3 shows the K and Kh values for all the submitted runs. LR indicates logistic regression with linear transfer function; log_LR indicates logarithmic transfer function; avg_LR indicates the average of the two. B indicates the number of documents in the boolean base run, while the constants 25000 and 100000 indicate that these values were fixed for all topics. In all cases we chose Kh (the value of K for highly relevant documents) to be K/2.

wat2text satisfies the TREC 2008 requirement that one run be derived exclusively from the request_text field, while wat3nobool excludes all documents in the supplied list for the purpose of enhancing the judging pool.

Runs	K	Kh
wat1fuse	avg_LR	K/2
wat2text	25000	12500
wat3nobool	100000	50000
wat4fuse	LR	K/2
wat5fuse	log_LR	K/2
wat6fuse	25000	12500
wat7fuse	100000	50000
wat8fuse	B	B/2

Table 3: K methods

3 Legal Track Results

Table 4 shows this year's main measures of $F1@K$ and $F1@R$ and last year's main measure of $R@B$. Because six of the runs are identical except for different K values they have the same $F1@R$ and $R@B$ values. It is very disappointing that wat7fuse has the highest $F1@K$ because for this run K is set to 100000; the number of returned documents. This indicates our methods of optimizing K decreases system performance.

Table 5 shows the mean average precision (MAP), bpref scores and the number of relevant documents returned for our legal track runs. A point of interest is

run	$F1@K$	$F1@R$	$R@B$
wat1fuse	0.1296	0.2427	0.3289
wat2text	0.1669	0.2306	0.2464
wat3nobool	0.1569	0.1744	0.1944
wat4fuse	0.1538	0.2427	0.3289
wat5fuse	0.0532	0.2427	0.3289
wat6fuse	0.1747	0.2427	0.3289
wat7fuse	0.2204	0.2427	0.3289
wat8fuse	0.2005	0.2427	0.3289

Table 4: Legal Track Results

that wat3nobool found 1174 relevant documents. The number of relevant documents found by the TREC provided boolean run is 2072. Also, the total number of relevant for this set of topics is 3564. This result indicates a vast numbers of relevant documents not returned by the boolean query. It also shows this method is good at finding them.

run	map	bpref	# relevant
wat1fuse	0.1459	0.5542	3153
wat2text	0.1049	0.4821	2916
wat3nobool	0.0366	0.2118	1174
wat4fuse	0.1459	0.5542	3153
wat5fuse	0.1459	0.5542	3153
wat6fuse	0.1459	0.5542	3153
wat7fuse	0.1459	0.5542	3153
wat8fuse	0.1459	0.5542	3153

Table 5: Classic Measure Results

We spent no time optimizing for Kh as we had no training data. For all runs we set Kh equal to half of K. Table 6 show the $F1@K$ and $F1@Kh$ results for the submitted results. It is again disappointing that a constant($Kh = 12500$) is the top performing run.

run	$F1@K$	$F1@Kh$
wat1fuse	0.1296	0.0934
wat2text	0.1669	0.0980
wat3nobool	0.1569	0.0770
wat4fuse	0.1538	0.1063
wat5fuse	0.0532	0.0336
wat6fuse	0.1747	0.1064
wat7fuse	0.2204	0.0998
wat8fuse	0.2005	0.1047

Table 6: Highly Relevant Results

4 Discussion

We learned little about legal IR from our TREC 2008 efforts. In effect, our entire effort was devoted to "gaming" the evaluation method in two ways: first, to guess the optimal value of K for an evaluation measure heavily influenced by this guess; second to run up the number of amenable documents in the pool by submitting a run excluding the boolean results.

Using the main metric of F1@K our best preforming run was wat7fuse with a score of 0.2204. A constant K value of 100000 is used in the wat7fuse run. We believe our best run would be wat1fuse but F2@K is only 0.1296 Performance is significantly hurt by our linear regression learning method to find the optimal K value.

The wat3nobool run is very interesting. It finds 1174 of the 1492(79%) found relevant documents not contained in the boolean run. More study is needed to determine what such a different run has on judgement pool.

References

[BKFS95] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manage.*, 31(3):431–448, 1995.

[CC96] C.L.A. Clarke and G.V. Cormack. Interactive substring retrieval (MultiText Experiments for TREC-5). In *5th Text REtrieval Conference*, Gaithersburg, MD, 1996.

[CCKL00] C. L. A. Clarke, G. V. Cormack, D. I. E. Kisman, and T. R. Lynam. Question answering by passage selection. In *9th Text REtrieval Conference*, Gaithersburg, MD, 2000.

[CCL01] Charles L. A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. Exploiting redundancy in question answering. In *SIGIR Conference 2001*, New Oreans, Louisiana, 2001.

[CL07] G.V. Cormack and T.R. Lynam. MultiText Legal Experiments at TREC2007. In *2007 Text REtrieval Conference*, Gaithersburg, MD, 2007.

[LBCC04] Thomas R. Lynam, Chris Buckley, Charles L. A. Clarke, and Gordon V. Cormack. A multi-system analysis of document and term selection for blind feedback. In *CIKM '04: Thirteenth ACM conference on Information and knowledge management*, pages 261–269, 2004.

[LC06] Thomas R. Lynam and Gordon V. Cormack. On-line spam filter fusion. In *29th ACM SIGIR Conference on Research and Development on Information Retrieval*, Seattle, 2006.

[RWJ+95] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Third Text REtrieval Conference*, Gaithersburg, MD, 1995.

[SF94] Joseph A. Shaw and Edward A. Fox. Combination of multiple searches. In *Text REtrieval Conference*, pages 0–, 1994.

[Tom07] S. Tomlinson. Overview of the TREC 2007 Legal Track. In *2007 Text REtrieval Conference*, Gaithersburg, MD, 2007.