

# **Index Pruning and Result Reranking: Impact on Ad-Hoc Retrieval and Named Page Finding**

Stefan Büttcher

Charles L. A. Clarke

Peter C. K. Yeung



TREC 2006 Terabyte Track  
November 16, 2006

# Motivation

The Terabyte track is comprised of 3 subtasks:

1. ad-hoc retrieval
2. named page finding
3. efficiency

The queries used in the efficiency task contain the queries from the ad-hoc and the named page finding task, but without any special annotation.

How does the search engine know the type of a given search query?

“blue grass music festival history”

“arizona retirement system history”

“kalamazoo public library history”

*Does it have to know the type?*

# Evaluation Methodology

A unified measure for the search quality in ad-hoc retrieval and named page finding does not exist.

To be able to evaluate both query types at the same time, we define a new measure *Goodness@k* ( $G@k$ ):

- for ad-hoc retrieval:  $Goodness@k = Precision@k$ ;
- for named page finding:  $Goodness@k = Success@k$ .

Our standard measure is  $G@10$ .

This is completely arbitrary, but as good as any other evaluation measure.

# Retrieval Baseline

Our retrieval baseline is Okapi BM25:

$$S_Q(D) = \sum_{q \in Q} w_q * \frac{(k_1 + 1) * f_{D,q}}{f_{D,q} + k_1 * (1 - b + b * dl / avgdl)}$$

$D$ : document

$Q$ : query

$q$ : query term

$f_{D,q}$ : frequency of  $q$  within  $D$

$dl$ : document length of  $D$  (number of tokens)

$avgdl$ : average document length in the collection

All queries are stemmed using Porter's algorithm.

# Initial Parameter Tuning

BM25 parameter tuning (finding the optimal value for the document length normalization parameter  $b$ ):

Topics	$b = 0.60$	$b = 0.75$	$b = 0.90$
701-750 (ad-hoc '04)	0.5204	0.5041	<b>0.4612</b>
751-800 (ad-hoc '05)	<b>0.6280</b>	0.5920	<b>0.5620</b>
801-850 (ad-hoc '06)	0.5240	0.5200	<b>0.4840</b>
601-872 (NP '05)	0.5040	0.5159	0.5119
901-1081 (NP '06)	<b>0.4641</b>	0.5028	0.5249

Evaluation measure: Goodness@10.

When increasing  $b$  from 0.6 to 0.9:

- ad-hoc retrieval: -11% ('04), -11% ('05), -8% ('06);
- named page finding: +2% ('05), +13% ('06).

*Bold numbers are statistically significant ( $p < 0.05$ ) compared to the default value  $b=0.75$ .*

# Initial Parameter Tuning

Why does changing the document length normalization parameter  $b$  affect the search quality for the two query types differently?

Decreasing  $b$  gives higher scores to long documents.

Hypothesis: Relevant documents are longer for the ad-hoc topics than for the named page finding topics.

A quick look into the GOV2 collection:

- ad-hoc topics 751-800 (TB '05)  
average relevant document contains 6,537 tokens
- named page finding topics 601-872 (TB '05)  
average relevant document contains 2,350 tokens

# Document Structure

Original BM25, without taking document structure into account:

Topics	$b = 0.60$	$b = 0.75$	$b = 0.90$
701-750 (ad-hoc '04)	0.5204	0.5041	<b>0.4612</b>
751-800 (ad-hoc '05)	<b>0.6280</b>	0.5920	<b>0.5620</b>
801-850 (ad-hoc '06)	0.5240	0.5200	<b>0.4840</b>
601-872 (NP '05)	0.5040	0.5159	0.5119
901-1081 (NP '06)	<b>0.4641</b>	0.5028	0.5249

Variant of BM25F, giving special weight to document title and markup (<b>, <i>, ...),  $b = 0.75$ :

Topics	G@3	G@10	MAP	MRR
701-750 (ad-hoc '04)	0.5442	0.4980	0.2373	0.7398
751-800 (ad-hoc '05)	0.6667	0.5900	0.3065	0.7895
801-850 (ad-hoc '06)	0.5200	0.4880	0.2564	0.3303
601-872 (NP '05)	0.4683	0.5794	n/a	0.4236
901-1081 (NP '06)	0.3923	0.5193	n/a	0.3528

# Static Index Pruning

## Last year:

- Term-centric static index pruning (Carmel's method).

## This year:

- Document-centric static index pruning, based on Kullback-Leibler divergence (KLD).

## General idea:

- Create a pruned index that is small enough to fit into main memory.
- Process queries using the pruned in-memory index.
- Only access the unpruned on-disk index when a query term has no entry in the pruned index.



# Static Index Pruning

## KLD-based document-centric static index pruning

The KLD between two discrete probability distributions  $P, Q$  is:

$$KLD(P, Q) = \sum_x P(x) * \log\left(\frac{P(x)}{Q(x)}\right)$$

Let  $P$ : unigram term distribution for document  $D$ ;

$Q$ : unigram term distribution for text collection  $C$ .

Then  $KLD(P, Q)$  is a measure for how different  $D$  is from the whole collection  $C$ .

*Assumption:*  $D$ 's most representative terms are those that make the greatest contribution to  $KLD(P, Q)$ . – idea stolen from Carpineto et al. (2001)

**Pruning method:** Pick the top  $p\%$  terms from each document, according to their contribution to  $KLD(P, Q)$ ; discard the rest.

University of

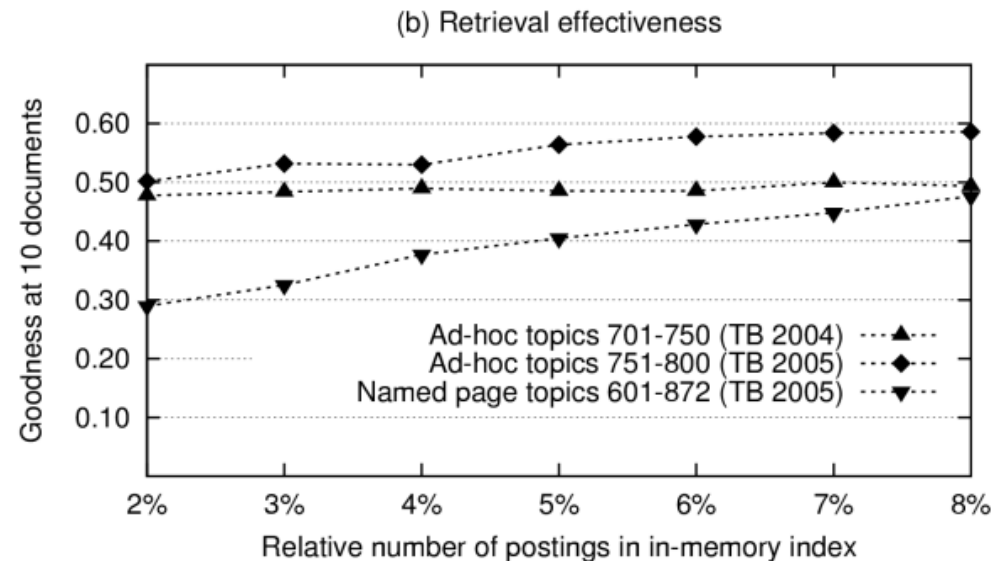
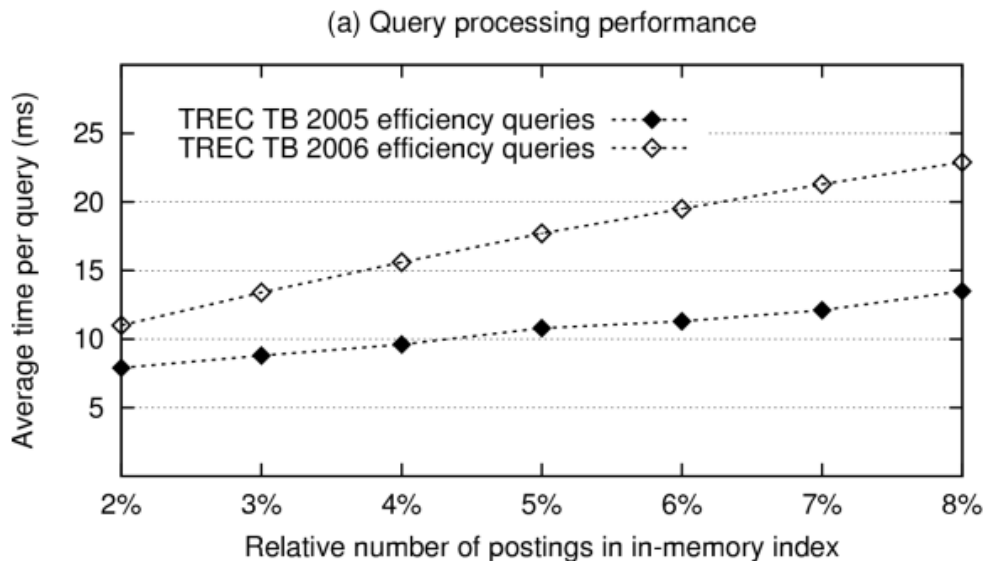
**Waterloo**



# Static Index Pruning

Effect of document-centric static index pruning on:

- retrieval efficiency (avg. time per query);
- search quality (Goodness@10).



Search quality baseline was:

0.498 (ad-hoc '04), 0.590 (ad-hoc '05), 0.579 (NP '05).

# Static Index Pruning

Unpruned index (retrieval baseline):

Topics	G@3	G@10	MAP	MRR
701-750 (ad-hoc '04)	0.5442	0.4980	0.2373	0.7398
751-800 (ad-hoc '05)	0.6667	0.5900	0.3065	0.7895
801-850 (ad-hoc '06)	0.5200	0.4880	0.2564	0.3303
601-872 (NP '05)	0.4683	0.5794	n/a	0.4236
901-1081 (NP '06)	0.3923	0.5193	n/a	0.3528

Document-centric index pruning, taking the top 5% terms from each document:

Topics	Latency	G@3	G@10	MRR
701-750 (ad-hoc '04)	27.2 ms	0.5510	0.4857	0.720
751-800 (ad-hoc '05)	22.2 ms	0.6400	0.5640	0.770
801-850 (ad-hoc '06)	24.1 ms	0.4933	0.5000	0.608
601-872 (NP '05)	33.9 ms	0.2976	0.4048	0.282
901-1081 (NP '06)	28.0 ms	0.3260	0.4088	0.290

# Reranking Based on Language Models

Very similar to Lavrenko's models of relevance:

Build a unigram language model  $M_{rel}$  from the text found in the top  $k$  documents retrieved in an initial retrieval stage (BM25F).

Rerank the top 1,000 documents by adjusting their score:

$$S_{new}(D) := S_{old}(D) - \rho * \text{KLD}(M_D, M_{rel}),$$

where  $M_D$  is  $D$ 's language model.

We experimented with two variants:

- $\rho = 1$  (because it is the simplest possible incarnation);
- $\rho = |Q|$ , where  $|Q|$  is the number of query terms, taking into account that  $S_{old}(D)$  is linear in  $|Q|$ .



# Reranking Based on Language Models

Impact on ad-hoc retrieval for  $\rho = 0$  (left), 1 (middle),  $|Q|$  (right):

Topics	P@10	P@20	MAP	bpref
701-750 (ad-hoc '04)	0.4980/0.5367/0.5531	0.4745/0.5214/0.5173	0.2373/0.2592/0.2467	0.3176/0.3359/0.3379
751-800 (ad-hoc '05)	0.5900/0.6320/0.6320	0.5440/0.5900/0.6270	0.3065/0.3361/0.3291	0.3632/0.3898/0.4133
801-850 (ad-hoc '06)	0.4880/0.5240/0.5500	0.4310/0.5030/0.5220	0.2564/0.2948/0.2791	0.3303/0.3561/0.3734

Reranking based on estimated models of relevance increases search quality substantially:

- P@10 increases between 7% ('05) and 13% ('06);
- P@20 increases between 9% ('04) and 21% ('06);
- bpref increases between 6% ('04) and 14% ('05).

# Reranking Based on Language Models

Effect of reranking on ad-hoc retrieval and named page finding.  
Search quality measured by Goodness@10:

Topics	$\rho = 0$	$\rho = 1$	$\rho =  Q $
701-750 (ad-hoc '04)	0.4980	<b>0.5367</b>	<b>0.5531</b>
751-800 (ad-hoc '05)	0.5900	<b>0.6320</b>	0.6320
801-850 (ad-hoc '06)	0.4800	<b>0.5240</b>	0.5500
601-872 (NP '05)	0.5794	0.5754	<b>0.3929</b>
901-1081 (NP '06)	0.5193	0.5193	<b>0.3315</b>

Bold numbers indicate statistical significance ( $p < 0.05$ ) compared to the baseline ( $\rho = 0$ ).

⇒ *This method helps for ad-hoc retrieval, but is devastating for named page finding.*

# Reranking Based on Anchor Text

The lazy man's approach to named page finding:

1. Perform a standard retrieval run (BM25F).
2. Adjust document scores based on anchor text found in top documents retrieved.

Easier than PageRank. Hopefully works as well.

## General idea:

- For each document  $D$ , build a pseudo-document  $D'$  that is the concatenation of the anchor text associated with hyperlinks pointing to  $D$ .
- Weight term occurrences in the pseudo-document based on the original score of the source document.
- Update  $D$ 's score:  $S_{new}(D) := S_{old}(D) * (1 + \rho * S_{BM25}(D'))$ .

University of

**Waterloo**



*Details are in the paper.*

**Index Pruning and Result Reranking  
Effects on Ad-Hoc Retrieval and Named Page Finding**

Stefan Buettcher  
<sbuettch@uwaterloo.ca>

# Reranking Based on Anchor Text

Impact on search quality for named page finding (for  $\rho=0.2$ ):

Topics	MRR	S@3	S@10
NP601-872	0.424/0.459	0.468/0.508	0.579/0.611
NP901-1081	0.353/0.419	0.392/0.508	0.519/0.569

Impact on Goodness@3 for ad-hoc retrieval and NP finding (bold numbers indicate statistically significant difference from baseline):

Topics	$\rho = 0$	$\rho = .1$	$\rho = .2$	$\rho = .3$
701-750 (ad-hoc '04)	0.544	0.503	0.476	<b>0.463</b>
751-800 (ad-hoc '05)	0.667	<b>0.607</b>	<b>0.600</b>	<b>0.593</b>
801-850 (ad-hoc '06)	0.520	0.520	0.513	0.507
601-872 (NP '05)	0.468	<b>0.496</b>	<b>0.508</b>	<b>0.516</b>
901-1081 (NP '06)	0.392	<b>0.470</b>	<b>0.508</b>	<b>0.492</b>



# Conclusion

We have investigated three different techniques:

- static index pruning (document-centric, using KLD);
- reranking based on relevance models (like Lavrenko, KLD);
- reranking based on anchor text.

The three techniques have different impact on ad-hoc retrieval and named page finding:

- static index pruning works well for ad-hoc retrieval, but has a devastating effect on search quality for NP finding;
- reranking based on language models only works for ad-hoc retrieval;
- reranking based on anchor text only works for NP finding.

**Conclusion:** (pick one)

- The standard search quality measures do not reflect the true quality. We need better measures!
- Ad-hoc retrieval and named page finding are fundamentally different. How do we deal with that?