Probabilistic Static Pruning of Inverted Files

ROI BLANCO and ALVARO BARREIRO
University of A Coruña

Information retrieval (IR) systems typically compress their indexes in order to increase their efficiency. Static pruning is a form of lossy data compression: it removes from the index, data that is estimated to be the least important to retrieval performance, according to some criterion. Generally, pruning criteria are derived from term weighting functions, which assign weights to terms according to their contribution to a document's contents. Usually, document-term occurrences that are assigned a low weight are ruled out from the index. The main assumption is that those entries contribute little to the document content.

We present a novel pruning technique that is based on a probabilistic model of IR. We employ the Probability Ranking Principle as a decision criterion over which posting list entries are to be pruned. The proposed approach requires the estimation of three probabilities, combining them in such a way that we gather all the necessary information to apply the aforementioned criterion.

We evaluate our proposed pruning technique on five TREC collections and various retrieval tasks, and show that in almost every situation it outperforms the state of the art in index pruning. The main contribution of this work is proposing a pruning technique that stems directly from the same source as probabilistic retrieval models, and hence is independent of the final model used for retrieval.

Categories and Subject Descriptors: H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval—*Retrieval models*; H.3.4 [**Information Storage And Retrieval**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Pruning, inverted files, probabilistic models, compression, efficiency

ACM Reference Format:

Blanco, R. and Barreiro, A. 2010. Probabilistic static pruning of inverted files. ACM Trans. Inform. Syst. 28, 1, Article 1 (January 2010), 33 pages.
DOI = 10.1145/1658377.1658378 http://doi.acm.org/10.1145/1658377.1658378

This research was co-funded by FEDER, Ministerio de Ciencia e Innovación and Xunta de Galicia under projects TIN2008-06566-C04-04/TIN and 07SIN005206PR.
Authors' address: IRLab, Computer Science Department, University of A Coruña, Spain; email: {rblanco, barreiro}@udc.es.

DOI 10.1145/1658377.1658378 http://doi.acm.org/10.1145/1658377.1658378

1. INTRODUCTION

Information retrieval (IR) systems address the problem of how to retrieve information in response to a user need or task at hand, from a given repository of information, such as a document collection. IR systems organize the document collection in an index (also known as inverted file), for efficient access. An index is a data structure that maps terms to the documents that contain them. The most efficient index structure for IR systems [Witten et al. 1999] is the inverted file, which records information on which terms appear in which documents and how often (posting lists). The relevance of a document to a query is estimated mainly using term weights, which capture the contribution of a term in a document. There exist different ways for computing term weights [van Rijsbergen 1979].

Today, the amount and type of information stored in the index is increasing exponentially. Additionally, the emergence of new environments for IR systems, for example mobile devices, desktop search in personal computers, or distributed and peer to peer IR, has produced new IR applications in which memory managing is crucial. This fast growth of indexable data forces modern IR systems to face new challenges: how and which data is stored and retrieved is an open problem, and there is a need for fast and effective systems that solve this issue. This article addresses the problem of efficient indexing through a compression mechanism, namely pruning. Index pruning consists of removing from the index data estimated to be the least important to retrieval performance, according to some criterion [Carmel et al. 2001b].

In this work, we propose an index pruning technique that uses a novel decision mechanism. Previous approaches to the problem are mostly fuelled by the fact that a user is only interested in a reduced number of top retrieved documents. Hence, based on the retrieval model employed by a particular IR system, pruning techniques decide which information is processed for answering users' queries. Our pruning technique is based on the Probability Ranking Principle [Robertson 1977], and is independent of the retrieval model in use.

The proposed technique requires the estimation of three different probabilities, and its outcome depends on the quality of these estimations and their combination. Therefore, as a side result, we propose novel techniques for these estimations, which are based on some well-stated facts in the IR field.

We evaluate our technique extensively, using five standard TREC [Voorhees and Harman 2005] collections of different size and genre, and assess it by queries of different size and nature. We compare the new pruning algorithm with a well-known baseline [Carmel et al. 2001b]. Experimental evaluation shows that our technique outperforms the baseline in terms of mean average precision, and performs better than, or at least comparably to, the baseline in terms of early precision. Also, we show that our proposed pruning method can result in improvements in retrieval precision over a full (not pruned) index, independently of the way parameters are tuned in the retrieval models.

The remainder of this article is organized as follows. Section 2 presents the background for index pruning, with emphasis on Carmel's pruning method, [Carmel et al. 2001b], which is used extensively in this work. Section 3 presents

the theoretical background of this work, and the derivation of our proposed pruning technique. Section 4 presents a way for estimating the probabilities involved in the pruning technique, and a way to obtain some recommended parameter values. Section 5 presents and discusses an experimental evaluation of our technique. Further discussion can be found in Section 6. Section 7 summarizes our findings and gives intended future work.

2. INDEX PRUNING

IR systems need to be efficient to keep low response times. Efficient performance is usually achieved by employing suitable data structures (inverted files [Witten et al. 1999]), policy-driven memory caches [Baeza-Yates et al. 2007], and by using adequate algorithms to access these data structures (e.g. posting file ordering [Anh and Moffat 2002]).

Index pruning can be dynamic or static. Dynamic index pruning aims to improve system efficiency by stopping the inverted file scanning during query time, when some criterion is satisfied. Some examples of dynamic pruning are the MAXSCORE optimization [Turtle and Flood 1995], or its more recent variation described in Strohman et al. [2005]. Anh and Moffat [2006] presented a different approach in which document pointers have to be sorted according to *impacts*. These impacts reflect the partial contribution of the term occurrence in a document, and provide fast index scanning. The work by Long and Suel [2003] addresses the issue of dynamic pruning when there is a global page ordering for documents, given by any query-independent document ranking metric (like Pagerank [Brin and Page 1998], for instance) and presents six heuristics to tackle this problem. The work by Theobald et al. [2004] presents a dynamic pruning method that preserves the top-k results (when using the pruned or unpruned index) using a probabilistic score prediction for each query term and efficiently managing priority queues. Dynamic pruning is a very efficient strategy for reducing answering times [Zobel and Moffat 2006] without compromising the retrieval performance.

Static pruning consists of compressing the index size, by pruning entries from the postings file, according to some criterion. Unlike dynamic pruning, this is done off-line and is query-independent. Static pruning is a *lossy* compression technique, because it is not possible to bring the compressed data back to its exact uncompressed form.

Static pruning has two advantages: it not only reduces query time, but also disk occupancy, and it is query- independent; hence it can be done off-line. The trade-off between efficiency and effectiveness is of outmost importance for static pruning approaches.

Static index pruning attempts to discard the least important data from the index. This data is estimated according to some relevance criteria [Carmel et al. 2001b]. A number of proposals have been made to address static pruning, which can fall into two different categories: term-based and document-based.

Document-based pruning discards the terms from a document that are less representative of the document's content. The main advantage of document-based pruning is that it can be applied on-the-fly while indexing the collection

```
Prune(k,ε)
1:  for every term t do
2:      {docs} = posting(t)
3:      if |{docs}| > k
4:          ∀D ∈ {docs}
5:              A(t, D) = score(t, D)
6:          z_t ← k-th highest entry in row t of A
7:          τ_t ← ε · z_t
8:          ∀D ∈ {docs}
9:              if A(t, D) < τ_t
10:                 remove entry D from posting(t)
```

Fig. 1. Carmel et al. [2001b] Top-k algorithm. The procedure takes two parameters, ϵ and z_t and discards the posting entries that score less than $\epsilon \cdot z_t$, where z_t is the k-th highest score in the whole posting.

[Büttcher and Clarke 2006]. However, on-the-fly pruning requires estimating collection statistics using only partial subcollections, which may not be accurate enough. Term-based pruning reduces the index size by discarding the term occurrences that are less likely to affect the retrieval performance of a specific retrieval model.

Index pruning is uniform when it is applied to all the terms or documents in the same way. A first detailed overview of static index pruning methods is given in Carmel et al. [2001b], where an experimental overview of uniform and term-based pruning is presented. The work presented in [Carmel et al. 2001b] forms the baseline of our work.

The algorithm of Carmel's method (see Figure 1) aims at preserving the top results when either the original or the pruned index is used by a retrieval system. It is an *idealized pruning algorithm*, which ensures the similarity of the top k returned results for queries of less than $r = 1/\epsilon$ terms, where k and ϵ are parameters. This property holds if the scores assigned by the IR system are the same before and after pruning. The procedure operates on a term-by-term fashion, selecting which term-document pairs are ruled out from the index on the basis of their scores (in the original paper TF-IDF). For every term in the dictionary, the algorithm computes the scores of the documents contained in its posting list: this reflects the partial contribution of the term in any query it appears in. Then, the method selects the k-th highest score, z_t, and sets $\tau_t = \epsilon * z_t$. Every document-term pair in that posting that scores lower than τ_t is discarded from the index.

Finally, although the algorithm preserves the similarity of the top k returned results, which is a nice theoretical property, the pruning levels achieved are negligible [Carmel et al. 2001b]. In order to obtain any significant index reduction, it is necessary to shift every term-document score in the index by subtracting the minimum score of the index from every document score. In practice, the pruning algorithm is applied after this ad hoc modification of the inverted file. This accomplishes excellent results, but the property of preserving the top-k results despite the index used (pruned or unpruned), is not guaranteed.

In this work we do not impose any similarity preserving conditions, as the retrieval quality is measured in terms of precision and not by document ranking

similarity. Hence, we omit the shifting step and increment the amount of pruning by setting higher ϵ thresholds.

Carmel's pruning method has been further investigated repeatedly [Carmel et al. 2001a; Büttcher and Clarke 2006; Blanco and Barreiro 2007a], and is considered state of the art in index pruning. Other work uses different ideas to tackle static pruning. Blanco and Barreiro [2007b] presented a comparison of four collection-dependent algorithms for pruning low-contribution terms and their impact on the efficiency and retrieval performance. Büttcher and Clarke [2006] proposed a document-centric pruning approach that uses a query expansion technique based on Kullback-Leibler (KL) divergence. Their algorithm produces excellent outcomes in terms of query processing speed gains. Results were reported on the .GOV2 collection, which is a 428 Gigabyte collection, and effectiveness figures were reported on 50 short queries. Another approach for performing document-based pruning is to replace the documents by their respective summaries. Summary indexing seems to improve precision while incurring a large recall loss [Brandow et al. 1995; Sakai and Sparck-Jones 2001]. The work by Ntoulas and Cho [2007] addresses the issue of resource handling for the pruning of terms and posting entries, by keeping the unpruned index on disk and determining the conditions and pruning level necessary to keep the top results the same, under a uniform pruning regime.

This work proposes a different approach, based solely on relevance estimations and not retrieval models. It is shown that our proposed algorithm is both theoretically sound and also beneficial to retrieval.

3. PROBABILITY RANKING PRINCIPLE FOR INDEX PRUNING

3.1 Derivation of Pruning Criterion from Probability Ranking Principle

In this section, we show how to formally derive a pruning mechanism based on the *Probability Ranking Principle* (PRP).

The Probability Ranking Principle (PRP) [Robertson 1977; van Rijsbergen 1979] states: "If a [...] retrieval system's response to each request is a ranking of the documents in the collections in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data."

PRP can be used for the retrieval of documents, and it is the foundation of the probabilistic model of IR. The fundamental idea behind this work is to introduce this principle for index pruning as opposed to document retrieval.

Given a document and a query, represented by the random variables D and Q, a probabilistic IR model calculates the probability of D being relevant $p(r|D, Q)$ and nonrelevant $p(\overline{r}|D, Q)$. It was shown in van Rijsbergen [1979] that the ranking produced by the odds-ratio of relevance and nonrelevance is equivalent to the ranking produced by PRP.

PRP can be seen as a Bayes decision rule considering the classes of relevance r and not relevance $\overline{r}$. The probability of a document being relevant (respectively

nonrelevant) to a query has an associated error (ξ). Formally,

$$p(\xi|D, Q) = \begin{cases} p(r|D, Q) & \text{if we decide } D \in \overline{r} \\ p(\overline{r}|D, Q) & \text{if we decide } D \in r. \end{cases} \quad (1)$$

Each type of error can be associated with a *cost*, so that our decision rule would minimise the *risk* of assigning a document to a wrong class. Consider the following definitions:

—let c_{rr} be the cost of deciding $D \in r$ when $D \in r$;
—let c_{rn} be the cost of deciding $D \in r$ when $D \in \overline{r}$;
—let c_{nn} be the cost of deciding $D \in \overline{r}$ when $D \in \overline{r}$;
—let c_{nr} be the cost of deciding $D \in \overline{r}$ when $D \in r$.

Under the risk-minimization perspective, the risk of classifying nonrelevant documents as relevant should be lower than the risk of classifying relevant documents as nonrelevant:

$$c_{rr}\, p(r|Q, D) + c_{rn}\, p(\overline{r}|Q, D) < c_{nr}\, p(r|Q, D) + c_{nn}\, p(\overline{r}|Q, D). \quad (2)$$

Or equivalently,

$$\frac{p(r|Q, D)}{p(\overline{r}|Q, D)} > \frac{c_{rn} - c_{nn}}{c_{nr} - c_{rr}}. \quad (3)$$

Equation (3) states that the ratio of the probability that a document is relevant to a query over the probability that a document is not relevant to a query should be greater than the cost ratio that acts as a threshold:

$$\epsilon = \frac{c_{rn} - c_{nn}}{c_{nr} - c_{rr}}. \quad (4)$$

The basic idea in this article is to consider every term in the lexicon as a single-term query. We are not interested in the ranking produced by PRP for this query, but only in identifying which document-query pairs satisfy Equation (3). If a document-query pair satisfies Equation (3) it is kept in the index, otherwise it is pruned. Therefore, Equation (3) acts as our pruning criterion where ϵ (Equation (4)) acts as a pruning threshold. More simply, this is a way of deciding, for every term-document occurrence, if it would be acceptable to consider that entry as relevant, given a query. In essence, the odds-ratio states how characterizing a term is in the context of a document.

The threshold ϵ can be set in several ways; some choices could be to estimate the collection difficulty or to use relevance information. In this work, we set the basic costs to $c_{rn} = c_{nr} = 1$, and $c_{nn} = c_{rr} = 0$, and hence the starting threshold is 1. This is a typical assumption in retrieval scenarios. However, as it will be shown in the experimental section, we increase this threshold in order to assess the pruning versus precision trade-offs.

For the estimation of the left hand side of Equation (3) we employ a decomposition of the odds-ratio of relevance and nonrelevance [Lafferty and Zhai 2003].

Applying Bayes' rule, this becomes:

$$\begin{aligned}\frac{p(r|Q,D)}{p(\overline{r}|Q,D)} &= \frac{p(D,Q|r)\ p(r)}{p(D,Q|\overline{r})p(\overline{r})} \\ &= \frac{p(Q|D,r)\ p(D|r)\ p(r)}{p(Q|D,\overline{r})\ p(D|\overline{r})\ p(\overline{r})} \\ &= \frac{p(Q|D,r)\ p(r|D)}{p(Q|D,\overline{r})\ p(\overline{r}|D)}. \end{aligned} \tag{5}$$

Equation (5) states that the ratio of the probability that a document is relevant to a query over the probability that a document is not relevant to a query can be decomposed into two odds: one query-dependent and one query-independent. Now assume that the document D is independent of the query Q under the conditioned event $\overline{r}$, $p(D, Q|\overline{r}) = p(D|\overline{r})\ p(Q|\overline{r})$. Therefore,

$$\frac{p(r|Q,D)}{p(\overline{r}|Q,D)} = \frac{p(Q|D,r)}{p(Q|\overline{r})} \cdot \frac{p(r|D)}{p(\overline{r}|D)}, \tag{6}$$

or under a binary term-independence assumption [Robertson et al. 1981] for a query of $|Q|$ terms:

$$\frac{p(r|Q,D)}{p(\overline{r}|Q,D)} = \prod_{i=1}^{|Q|} \left(\frac{p(q_i|D,r)}{p(q_i|\overline{r})} \right) \cdot \frac{p(r|D)}{1-p(r|D)}. \tag{7}$$

Equation (7) decomposes the ratio of the probability that a document is relevant to a query over the probability that a document is not relevant to a query into three distinct probabilities. The first probability, $p(q_i|D,r)$, is the query likelihood. The second probability, $p(r|D)$, is the prior document. The third probability, $p(q_i|\overline{r})$, is the probability of a term given nonrelevance. We show how we estimate each of these probabilities separately in Section 4.

In the formulation presented in Equation (7), it is assumed that query terms occur independently of each other (*term-independence assumption*). During the pruning stage we do not have any query information and it seems reasonable to follow the term-independence assumption. This makes sense because typically, scoring functions are additive, which means that the partial score that a term produces has nothing to do with the contribution of the other, and thus to the final score.

Our proposed pruning method is *uniform* because the threshold is the same for every term in the index. However, the probabilities to be estimated bias the amount of data to be pruned more towards some terms than others. This is a first procedural difference of this method from Carmel et al. [2001b]. The latter method decides the amount of pruning on the basis of the differences between the scores of a single term in different documents. A second difference is that the intention of this pruning mechanism is not to be faithful to the original ranking produced by a given scoring function, nor to make results imperceptible for a user, but to detect which terms are not significant in the context of a document, so those term occurrences can be discarded from the index. The pruning process is summarized in Figure 2.

```
Prune(ε)
1:  for every term q in the lexicon do
2:      {docs} = posting(q)
3:      p_q = p(q|r̄)
4:      ∀D ∈ {docs}
5:          p_d = p(r|D)
6:          p_r = p(q|D, r)
7:          s = (p_r/p_q) * (p_d/(1 − p_d))
8:          if s < ε
9:              remove D from {docs}
```

Fig. 2. PRP-based uniform pruning algorithm. The procedure takes a parameter ϵ that acts as a threshold. It calculates three probabilities and combines them in a score. Finally, it discards the posting entries that score less than the threshold.

In the framework presented here, it is possible to incorporate additional information about which terms are more likely to be pruned, or equivalently, less *relevant*, by altering the estimation of the probabilities in use.

Finally, it is worth stating that this algorithm is well-founded, because it uses PRP. If the conditions of PRP were to hold, indexes pruned using this technique would be optimal for retrieval tasks, (in the same way PRP is optimal for retrieval). In Section 4, we present each probability appearing in Equation (6), and show a feasible way of estimating them.

3.2 PRP and Probabilistic Retrieval Models

PRP [Robertson 1977; van Rijsbergen 1979] is the foundation of probabilistic retrieval models, and it has been stated as a criterion of optimum retrieval since the 70s. BIR (binary independence retrieval) is the probabilistic retrieval model that follows the term independence assumption and PRP [Robertson and Sparck-Jones 1976; van Rijsbergen 1977]. Robertson et al. [1981] developed a retrieval model based directly on the log-odds ratio and on following PRP. The relevance of a document to a query was estimated using within-document term frequencies, modeled with a 2-Poisson mixture distribution. The parameters of the distributions were estimated directly by the within-document frequency distributions, and its usage in weighting functions resulted in few performance benefits. Based on these same assumptions, Robertson and Walker [1994] developed a simpler model that incorporated some variables present in the 2-Poisson model in an ad hoc fashion. This model further resulted in the successful BM25 retrieval model. The work in Manmatha et al. [2001] considers a probabilistic model where the main focus is to calculate the probability of relevance and nonrelevance. The model employs the distribution of scores provided by a search engine in order to obtain the probability of relevance of a certain document. Assuming that different search engines provide independent rankings, those probabilities are combined. Probabilities and PRP allow for ranking fusion in a more principled way. Finally, an interesting connection between language and probabilistic models for IR using PRP was presented in Lafferty and Zhai [2003].

4. PROBABILITY ESTIMATIONS

Generally, statistical methods rely on the goodness of their estimations. In this section, we present how we estimate each component of Equation (6), namely,

—the probability of a term given a document and relevance, or query likelihood (Section 4.1),

—the prior probability of relevance of a document (Section 4.2),

—the probability of a term given non-relevance (Section 4.3).

We present and discuss each estimation separately. The estimations proposed are not the only ones possible; however, they turned out to work very well for pruning purposes.

It is important that the estimates of the probabilities combine well with each other. Weighting schemes make explicit assumptions for assigning term weights like doing logarithmic transformations and smoothing probabilities in a particular way. Those assumptions do not affect the ranking process, because the final value of the score is taken to be meaningless in most cases: ranking only needs a final ordering of the documents. The scenario presented here implies a more complex modelling approach, as the probabilities to be estimated should combine well with each other (considering smoothed estimations).

4.1 Query Likelihood

In Equations (3), (4), and (6), we presented our pruning criterion, which consists of the combination of three probabilities and a threshold. The first of these probabilities is $p(Q|D,r)$, which is the probability of a document generating a given query. This probability corresponds to the well-known query likelihood probability, a fundamental part of the language modelling approach [Zhai and Lafferty 2004].

The query likelihood probability refers to the probability of a document generating a given term q_i. This probability has to be smoothed for being of real use in retrieval. Some well known smoothing methods are Dirichlet prior and the linear interpolation scheme (or Jelinek-Mercer (JM) smoothing) [Zhai and Lafferty 2004]. The performance of smoothing methods is dependent on some particular parameters. In this work we chose JM smoothing, because of its relative stability for short queries. Hence, for every term q_i, $p(Q|D,r)$ is approximated by $p(q_i|D)$, as follows:

$$p(q_i|D) = (1-\lambda)p_{mle}(q_i|D) + \lambda\ p(q_i|\mathcal{C}),\ \lambda \in [0,1], \tag{8}$$

where p_{mle} is the maximum likelihood estimator for a term q_i, given a document D, $p_{mle}(q_i|d) = \frac{tf(q_i,d)}{|D|}$, tf is the frequency of a term in a document, $|D| = \sum_{w_i \in D} tf(w_i, D)$ (the document length), and λ is a parameter (we set λ to 0.6 in every collection; see Section 5.1).

$p(q_i|\mathcal{C})$ is also the maximum likelihood estimator for a term q_i, given a collection $\mathcal{C}$:

$$p(q_i|\mathcal{C}) = \frac{count(q_i,\mathcal{C})}{\sum_{q_j \in \mathcal{C}} count(q_j,\mathcal{C})}, \tag{9}$$

where $count(q_i,\mathcal{C})$ is the number of occurrences of q_i in collection $\mathcal{C}$.

In language models, the query likelihood component represents the probability of a document model generating a string of text (query): $p(Q|D)$. We approximate the probability of the document generating the query given relevance $p(Q|D,r)$ with a simpler probability estimate. A possible alternative way to tackle this issue is to use relevance-based language models [Lavrenko and Croft 2001], which capture the notion of relevance in the language modelling framework. In practice, relevance LMs can be seen as a way of incorporating pseudo-relevance feedback to the language modelling approach in a principled way. A relevance language model would incorporate estimation information about terms related to the query (in our case, a single term). We did not follow that path in this work, due to the ease of computing Equation (8), and the high performance it brings on its own for pruning.

4.2 Document Prior

The second of these probabilities to be estimated is $p(r|D)$, which is the prior probability of relevance of a document, without any query information. This corresponds to the likelihood that the document is relevant just by query-independent evidence. A document prior can be based on certain knowledge on the document structure or content, and this is currently an active area of research. For instance, in Web IR, the typical sources for document priors information come from the link structure of Web pages [Kraaij et al. 2002], such as the number of incoming links (in-links) and URL depth [Westerveld et al. 2002], or even the Pagerank [Brin and Page 1998] algorithm for ranking Web documents according to their popularity [Upstill et al. 2003].

In most cases $p(r|D)$ is taken to be uniform [Zhai and Lafferty 2004], therefore it has no effect on retrieval. However, there have been several studies where the document length and link structure have been encoded as a prior probability, for ad hoc and some non-ad-hoc tasks [Kraaij et al. 2002; Westerveld et al. 2002]. Overall, incorporating prior knowledge on documents into retrieval has been particularly effective for Web retrieval, namely *homepage* and *named page finding*, which refer to the retrieval of a single Web page.

Using the formulation presented here, even a uniform document prior would have an effect on pruning. The only way of avoiding the prior effect over the pruning algorithm is to set $p(r|D) = p(\overline{r}|D)$ so that the document prior component cancels out.

We further explored the issue of estimating this probability and developed a way for estimating the prior that worked well for every collection tested in the pruning framework. In this work $p(r|D)$ is estimated solely as a function of document length and without any linkage structure information; therefore our findings can also be applied to collections without link information.

The derivation of the length-based document prior is as follows. A previous approach for deriving a prior based on document length, considered $p(r|D)$ as a linear function on the number of tokens of a document, as it is supposedly reminiscent of the true shape of relevance [Singhal et al. 1996]. The work by Kraaij et al. [2002] models the prior probability and document length dependency

as a straight line, which tries to reflect this fact. The idea behind this prior is that longer documents are more likely to embody more topics, and hence should receive a higher prior probability. This agrees with the scope hypothesis [Robertson and Walker 1994], which states that longer documents cover more topics than shorter ones. On the other hand, the verbosity hypothesis states that longer documents cover the same number of topics than shorter ones, but they just use more words to do it. Some effective retrieval models (like BM25 for instance) are also based on a parameterized combination of both hypotheses. The detailed analysis of the true shape of relevance by Singhal et al. [1996] shows that a sigmoid (pseudo-linear) function fits better the relevance curve. This also allows for incorporating more elements into the modelling, as shown next.

The hyperbolic tangent function is one of the hyperbolic transcendental functions and a member of the sigmoid family. It has some nice properties we shall exploit, like

—$tanh(0) = 0$;
—its inflexion point is $x = 0$;
—$tanh(x) \in [-1, 1]$.

We employ linear transformations (in the form of $ax + b$) to adjust the hyperbolic tangent:

$$S(x) = a + b \cdot tanh(c \cdot (x - d)). \tag{10}$$

It is possible to model the behavior of the sigmoid function $S(x)$ using four parameters, a, b, c, and d. In particular, the slope is controlled with c, d is the inflexion point, and its bounding interval is determined by a and b.

These parameters should be adjusted in order to transform Equation (10) depending on some particular assumptions, with the goal of assigning a prior probability -$S(x)$- to every document based on its length -x-. A different set of assumptions/intuitions on how the document prior probability should behave may lead to interpretations different than the one presented next.

First of all, we take the center point to be the average document length $\overline{X_d}$. This decision is in accord with the study by Singhal et al. [1996], where the relation between document length and relevance followed a sigmoid function centered over the mean document length. The shape of the curve implies there are some bounding values x_1, x_2 in the x-axis ($x_1 < \overline{X_d}, x_2 > \overline{X_d}$) that decide for which document lengths the normalization is quasilinear and for which it is quasiuniform. This would be the same as considering $if\ x < x_1 \Rightarrow S(x) \approx S(x_1)$ and $if\ x > x_2 \Rightarrow S(x) \approx S(x_2)$.

Now, let $S(x_1) = lo$ and $S(x_2) = hi$, and let μ be the point where we consider the tangent reaches its maximum, $tanh(\mu) \approx 1$ (we take $\mu \approx 2$), then the curve family results in:

$$p(r|D) \approx S(dl) = \frac{hi - lo}{2} \cdot tanh\left(\frac{\mu \cdot (2dl - (x_1 + x_2))}{x_2 - x_1}\right) + \frac{hi + lo}{2}. \tag{11}$$

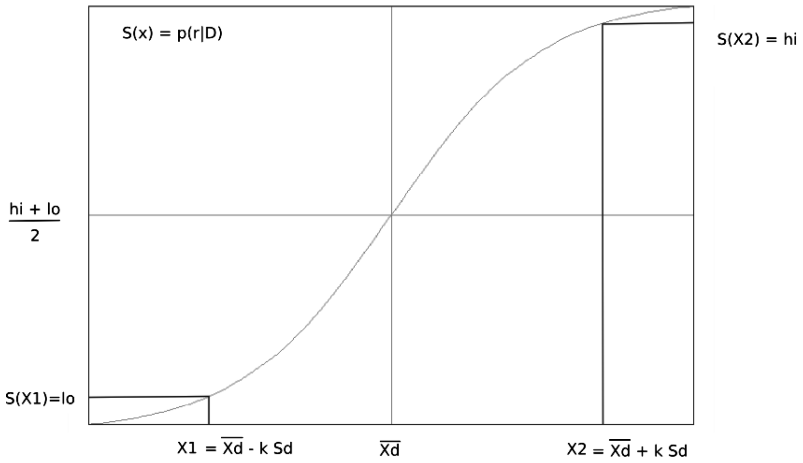

Fig. 3. $p(r|D)$ estimation using document lengths and a sigmoid function. $\overline{X_d}$ is the average document length, S_d the typical deviation over the document lengths, and hi, lo, and k are set to constant values.

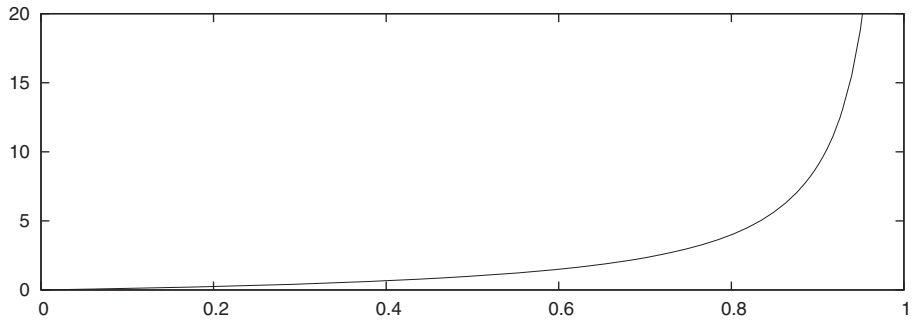

Fig. 4. $f(x) = x/(1-x)$.

In particular, the sigmoid function is useful for bounding both the effect of the prior on the pruning score and the range of document lengths the quasilinear effect is applied to. In the following paragraphs we model the dependency of $p(r|D)$ by just using document length and with the shape of a sigmoid function (Figure (3)).

First, we set the bounds $[lo, hi]$ for the contribution of the document prior probability to the right-hand side of Equation (7): $p(r|D)/(1 - p(r|D))$. Then, we discuss the document length range that the linear effect will be applied to.

Figure 4 is a plot of the $x/(1-x)$ function. In our case, this function reflects the contribution of the document prior to the pruning score (y-axis), given a prior probability $p(r|D)$ (x-axis). Moving to the right side of the x-axis implies a greater contribution if that value is accepted as a probability estimation. After a certain point, the growth of the function is exponential. Moving to the left side of the x-axis results in a very low contribution to the pruning score (Equation (7)), therefore its effect on discriminating between documents diminishes. In other words, this plot reveals some interesting facts about the bounds of the prior:

priors on the right of the x-axis will affect the pruning score more, and priors on the left side will result in a uniform effect for all the documents, regardless of their length. For these reasons, a reasonable interval for the prior (in this particular application) could be around 0.5, as this is the point where the prior contribution is 1: a prior value higher than 0.5 boosts the query-independent score, while it is decreased by values lower than 0.5. In this work we narrow the prior to the [0.4, 0.6] interval, taking a conservative approach.

Now that those bounds are set, we determine the document length range that produces a quasilinear behavior of $p(r|D)$ in the $[lo, hi]$ ([0.4, 0.6]) range, and asymptotical outside that interval. We relate that range to the standard deviation of the document lengths over a whole collection. More precisely, Equation (12) determines which fraction of the documents are going to be affected linearly, ruled by parameter k. Chebyshev's inequality states that in any data sample or probability distribution, nearly all the values are close to the mean value, and it provides a quantitative description of"nearly all" and "close to" in terms of the typical deviation. More formally, let X be a random variable with mean $\overline{X}$ and variance S^2, then

$$P(|X - \overline{X}| > k \cdot S) \leq \frac{1}{k^2}. \tag{12}$$

Figure 3 is the sigmoid function, as presented here: it grows linearly between a certain range, and it is possible to determine bounds for both axes. In the figure, hi and lo stand respectively for the maximum and minimum values of the prior estimation $p(r|D)$: 0.6 and 0.4, decided after the illustration of its effect on the pruning score, shown in Figure 4. The document length range affected is set by an interval around the average document length $\overline{X_d}$, with the help of Equation (12). The size of the interval is $2k$ times the typical deviation S_d over the document lengths: $[\overline{X_d} - kS_d, \overline{X_d} + kS_d]$. Thus, $x_1 = \overline{X_d} - kS_d$, $x_2 = \overline{X_d} + kS_d$ and k is set empirically to 2.

Substituting those values in Equation (11), the final equation governing the S-shaped document prior depends on the document length dl in the following way:

$$p(r|D) \approx S(dl) = \frac{1}{2} + tanh\left(\frac{dl - \overline{X_d}}{S_d}\right) \cdot \frac{1}{10}. \tag{13}$$

It is worth pointing out that this prior can be tweaked in many different ways, and some parameters can be tuned for specific collections, and thus it is possible to increase its overall performance. However, in all the experimental results presented in Section 5 the prior has been calculated with Equation (13), which is only dependent on collection statistics, for every collection tested.

4.3 Probability of a Term Given Nonrelevance

The last component needed to estimate in Equation (7) is the probability of seeing a term given *nonrelevance* $p(q_i|\overline{r})$. This is the only document-independent probability of our technique. Even though the pruning algorithm sets the same pruning threshold for every term, this probability biases the amount of

pruning towards some terms more than others. Recall Equation (6): document-dependent probabilities combined should score less than $\epsilon \cdot p(q_i|\overline{r})$ to be pruned. In this particular case, this probability determines how likely the term is to be pruned, when compared to document-dependent factors (query likelihood and document prior).

Traditional estimations of this probability [Croft and Harper 1979] consider the whole document set as nonrelevant to a query term, and so the probability of a term being nonrelevant would be:

$$p(q_i|\overline{r}) = \frac{df(q_i)}{N}\ , \tag{14}$$

where $df(q_i)$ is the number of documents q_i appears in, and N the total number of documents in the collection. This is the well-known inverse document frequency (idf) [Robertson and Sparck-Jones 1976] formulation, which is an integral part of many retrieval models. Equation (14) turned out to be an overestimation for $p(q_i|\overline{r})$, and of little use in the formulation derived in this work. The estimation is too aggressive: it is higher than the two document-dependent probabilities combined, if they are estimated as presented in previous sections. The consequence is that even with a threshold ϵ of 1, the majority of the inverted file would be pruned.

Instead of deriving an alternative formulation for idf (like smoothing it, for instance), we follow a different path and consider the collection as a model of *nonrelevance*. In this case, the probability of a term being nonrelevant is modelled according to the probability of the collection language model of generating the term. The collection language model is the maximum likelihood estimator (MLE) of the probability of seeing a term in a collection. It is worth pointing out that the role of this probability for LMs is considered equivalent to the role of idf for classical retrieval models [Zhai and Lafferty 2004]; if it is calculated using Equation (9), where the event space is the total number of occurrences of a term in the collection, the probability of nonrelevance of a term would be to consider every occurrence in the corpus as nonrelevant (just like idf considers every document as nonrelevant). We refine this probability next.

$$p(q_i|\overline{r}) = p(q_i|\mathcal{C}). \tag{15}$$

If idf is seen as a probability, it refers to the random event of selecting a single document that contains the term q_i (the total event space is determined by the documents). On the other hand, for $p(q_i|\mathcal{C})$, the event is to select a random term occurrence that happens to be q_i (the total event space is determined by the total term occurrences).

This estimation turns out to be useful in the framework presented here, and gives good practical results. This good behavior is likely due to the query likelihood component already incorporating the collection MLE $p(q_i|\mathcal{C})$ into its model.

However, it is still appealing to relate the nonrelevance probability with document frequency. In particular, the probability should increase monotonically with respect to document frequency, like in the case of idf. How we incorporated this property into the estimation is presented next.

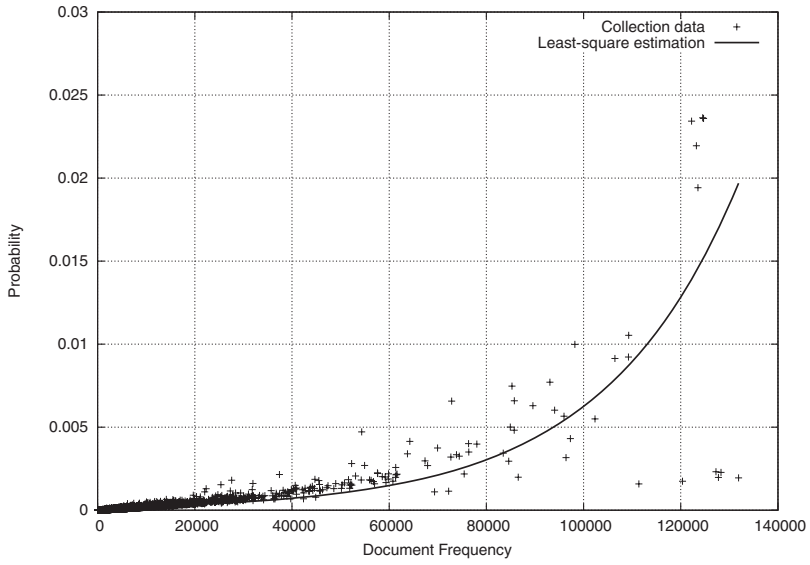

Fig. 5. $p(q_i|\mathcal{C})$ and estimated exponential fit (least-square), LATimes collection.

The points in Figure 5 represent the individual $p(q_i|\mathcal{C})$ values for every term in the LATimes collection. The probabilities (y-axis) are plotted against the term document frequency (x-axis). Data reveals that the probability grows exponentially after a certain document frequency level.

In order to correlate $p(q_i|\mathcal{C})$ with idf, we take the straight approximation of Equation (15) and fit the data to an exponential function with the form:

$$p(q_i|\overline{r}) = a \cdot e^{b \cdot x}. \tag{16}$$

In this case x corresponds to the document frequency of term q_i. We employ the well-known least squares technique in order to fit the data. It is the simplest form of regression, which provides a solution to the problem of how to find the best curve matching a given set of points, in our case $p(q_i|\mathcal{C})$. The procedure tries to minimize the sum of the squares of the residuals of the points of the curve, adjusting a and b iteratively.

For providing an estimate for the first iteration, we opted to calculate the initial values for the a and b parameters in Equation (16) by forcing the curve to pass through two points:

—the maximum value in the y-axis;

—(x_{av}, y_{av}), where x_{av} is the average value on the x-axis, and y_{av} is the average y-value on the $[x_{av} - 10000, x_{av} + 10000]$ interval. The choice of the width of this interval does not greatly affect the final parameter estimation and consequently, the shape of the exponential curve.

The least squares algorithm finds the best-fitting curve to a given set of points by minimizing the sum of the squares of the residuals (SSR) of the point from the curve. At each step (iteration) the procedure calculates the SSR, and then the parameter values for the next iteration. The process continues until

a stopping criterion is met: either the relative change in SSR is less than a convergence threshold (10^{-5} in this case), or it reaches a maximum number of iterations (100).

This fit is accurate and adequate for terms with low document frequency, but it is not completely adequate for some of the high document frequency terms, which appear at the rightmost extreme of Figure 5. The points below the solid line representing the fit are a few high-frequency terms, which only occur a few times in each document (tag words like *column* or *page* in a journalistic-domain collection or *http* in a Web collection); and the fit behaves well, boosting its probability of nonrelevance. On the other hand, the fit is not adequate for terms that appear several times in many documents. In IR such terms are called "stopwords" and are usually excluded from most computations, because they are of little significance to the final ranking of an IR system. Blanco and Barreiro [2007a] present an aggressive variation of Carmel's method, which prunes every term with document frequency $df > N/2$. This modification is simple and worked well in some test collections (retrieval-wise). In this work we follow a conservative approach and maintain the *fit* for terms with $df < N/2$ and rule out from the index the rest of them, that is equivalent to automatically assigning them a high probability of nonrelevance.

In this section we proposed one of several possible estimations for $p(q_i|\overline{r})$. It is possible to estimate this probability in other ways that use terms-specific information that is also document-independent. In Section 6 we assess the effect of removing stopwords for different pruning techniques.

5. EXPERIMENTS AND RESULTS

This section describes the experimental evaluation of our proposed probabilistic pruning method. The objective is to assess the trade-off between pruning level and retrieval precision. For detailed effects on the retrieval efficiency of pruning algorithms see Büttcher and Clarke [2006] and Carmel et al. [2001b].

This section is structured as follows: first we describe the experimental settings. Then, we define a baseline starting from Carmel's method and following some design considerations pointed out in Blanco and Barreiro [2007a]. Next, we compare the baseline and the proposed pruning model with three different retrieval models: TF-IDF (the model originally employed in Carmel et al. [2001b]), BM25 with standard and optimized settings (to avoid bias depending on parameter tuning) for every collection, and finally a parameter-free model based on divergence from randomness: DLHH [Amati 2006]

5.1 Experimental Settings

Retrieval performance has been assessed on five different TREC collections, which exhibit different characteristics. LATimes and TREC Disks 4 and 5 contain documents of single sources of news-press media, assumed to be fairly homogeneous. On the contrary, WT2G, WT10G, and .GOV contain crawls of Web pages, which come from a heterogeneous source (the World Wide Web). Overall, the collections vary in content, size, and statistics, so they form a

Table I. Collections and Topics

Collection	Size	Terms	Documents	Topics	Task
LATimes	450M	189,790	131,896	401–450	ad hoc
WT2G	2G	1,002,586	247,491	401–450	ad hoc
Disks 4&5	1.9G	521,469	528,155	301–450 + 601–700	ad hoc
WT10G	10G	3,140,837	1,692,096	451–550	ad hoc
.GOV	18.1G	2,788,457	1,247,753	1–225	NP & HP + TD

wide-ranged testbed for the pruning techniques. Evaluation has been done in two different ways:

—standard ad hoc retrieval with short and long queries;
—mixed Web track 2004 topics: topic distillation and homepage and namepage finding tasks.

Both types of experiments employ well-known TREC-style evaluation. This means that every collection has an associated set of queries and a human-assessed set of relevance judgements, relating which documents are relevant to those queries; this set of judgements is not complete across the whole set of documents. Standard TREC queries are formed of three types of fields, each describing the topic in more detail, namely *title*, *description*, and *narrative*. In this work we experiment with two types of queries for the ad hoc task: *short* (title only) and *long* (title and description). The Web track 2004 is formed of 225 short queries with different tasks (75 each) [Craswell and Hawking 2004]: homepage finding (the query is the name of a homepage Web site the user wants to reach), named page finding (the query is the name of a non-homepage Web site the user wants to reach) and topic distillation (queries describe general topics).

Retrieval performance is measured with mean average precision (MAP) and precision at 10 (P@10) [Baeza-Yates and Ribeiro-Neto 1999]. The latter measures the number of relevant documents retrieved from the top 10. This is generalizable to any P@k value. Let $P(R_k)$ be the precision obtained when R_k relevant documents have been retrieved. Then, MAP averages $P(R_k)$ over all the recall levels [Baeza-Yates and Ribeiro-Neto 1999].

The experimentation is designed in order to assess the effect of the pruning in retrieval. Results are reported for MAP and P@10 with respect to the percentage of postings kept in the inverted file (pruning level). Measuring the percentage of posting entries removed from the index turns out to be a good indicator of both final disk space savings in a compressed index and query performance gains [Blanco and Barreiro 2007b].

The evaluation is as follows. Given a collection, a judged query set and a retrieval model, first we evaluate the original unpruned index using the queries and relevance judgements. Then, we produce two sets of pruned indexes, one using Carmel's method (by varying the ϵ parameter) and the other with PRP-based pruning (varying the threshold). Every index in those sets is evaluated using the same conditions as the original unpruned index.

We employ three different retrieval models for the query-document ranking, two probabilistic and one vector-space based, which are presented next.

Weighting functions usually normalize the document length contribution to the final score by a factor that can be controlled by a parameter. Other models are built on some assumptions that permit surpassing the parameter dependency, and so they are parameter-free; for instance some models found in the DFR framework [Amati and van Rijsbergen 2002].

For each of the equations, we employ the following notation: Q is a query that contains $|Q|$ terms $\{q_1, q_2, \ldots, q_{|Q|}\}$, dl is the document length, $avgdl$ the average document length in the collection, N the number of documents in a collection, tf the term frequency inside the document, $avgtf$ the average term frequency in the document, TF the term frequency in the collection, df the document frequency of the term and qtf the term frequency in the query.

The first matching function considered, Equation (17), is a normalized TF-IDF variant as implemented in the SMART system [Buckley et al. 1995]:

$$sim(q,d) = \sum_{i=1}^{|Q|} \frac{\frac{\log(1+tf)}{\log(1+avgtf)} \cdot \log \frac{N}{df}}{\sqrt{(1-slope)\cdot avgdl + slope \cdot dl}}. \tag{17}$$

This matching function is based on the vector-space retrieval model by Salton et al. [1975]. The document-length normalization $\sqrt{(1-slope)\cdot avgdl + slope \cdot dl}$ normalizes document length so that there is no bias for longer documents. We use the pivoted normalization [Singhal et al. 1996], and the default value for *slope* is 0.2 [Buckley et al. 1995]. We use this function because it was the one employed in the first evaluation in Carmel et al. [2001b] as a core part of the JURU search engine Carmel et al. [2001a].

The second matching function is the probabilistic Okapi's Best Match25 (BM25) [Robertson et al. 1995], Equation (18). BM25 has been shown to be robust and stable in many IR studies.

$$sim(q,d) = \sum_{i=1}^{|Q|} \log \frac{N - df + 0.5}{df + 0.5} \frac{(k_1+1)\cdot tf}{K + tf} \frac{(k_3+1)qtf}{k_3 + qtf}, \tag{18}$$

where $K = k_1 \cdot ((1-b) + b \cdot \frac{dl}{avgdl})$. BM25 includes three parameters, namely k_1, k_3, and b. Some studies [Chowdhury et al. 2002; He and Ounis 2003], have shown that both k_1 and k_3 have little impact on retrieval performance compared to the b parameter. We set k_1 and k_3 to the constant values recommended in Robertson et al. [1995] ($k_1 = 1.2$, $k_3 = 1000$). The b parameter controls the document length normalization factor. The experiments have been carried out using two different settings:

—using the recommended value for $b = 0.75$ [Robertson et al. 1995];
—optimizing the value for b with 1D exploration ($b \in [0 \ldots 1]$, $b = 0.1, 0.2 \ldots$). For every pruning level, we optimize b for MAP and P@10, and we choose the values of b that maximize both measures.

DLHH [Amati 2006] is a recent probabilistic-based weighting model based on measuring the divergence between a random draw of occurrences of terms in documents and the actual distribution, Equation (19). It is based upon a hypergeometrical model, which embodies two probabilities: the relative

within-document term frequency and the entire collection term frequency. This model is useful because it is *parameter-free*, and therefore tuning is not necessary for obtaining high retrieval performance. We want to assess whether retrieval performance can benefit from pruning or not—if discarding information from an inverted file can result in increased retrieval performance. Parameter-free models reduce their number of variables and factors to take into account such a validation, and thus their convenience for this particular purpose.

$$sim(q,d) = \sum_{i=1}^{|Q|} \log\left(1+\frac{1}{tf}\right) \cdot tf \cdot \log\left(tf + \frac{avgdl}{dl} \cdot \frac{N}{TF}\right) + \frac{1}{2} \cdot \log\left(2\pi \cdot tf \cdot \left(1 - \frac{tf}{dl}\right)\right). \quad (19)$$

The parameter settings for the estimations of our proposed pruning algorithm described in Section 3 are set to the following values.

—To calculate the query-likelihood component $p(q_i|D,r)$, Section 4.1, λ in Equation (8) was set to 0.6. Preliminary experiments on some test collections demonstrated that setting this value performs well, although better performance can be obtained if it is tuned for a given dataset. However, we decided to skip any further tuning of λ in order to prove that the technique is robust.

—The document prior component $p(r|D)$, Section 4.2, is estimated by document length using Equation (13).

—The probability of a term given nonrelevance $p(q_i|\overline{r})$ is calculated by Equation (15) and interpolated with the algorithm described in Section 4.3. We effectively discard terms for which $df > N/2$ by assigning a high $p(q_i|\overline{r})$ value.

It is possible to obtain better results by tuning those parameters separately for each collection, but we omit those experiments to assess the behavior of the method using a default setup. The cost-associated threshold for the probabilistic-based pruning was initially set to 1. In the following figures the first point corresponds to the pruning level obtained with $\epsilon = 1$. The subsequent pruning levels were obtained by increasing that value.

Regarding the settings for Carmel's method, k was set to 10, as it is beneficial for P@10 [Blanco and Barreiro 2007a], and the subsequent pruning levels were obtained by modifying ϵ.

Next, we present a set of six different experiments. The first (Section 5.2) defines the baseline from a minor variant of Carmel's method, that considers the update of the statistics in the pruned index. In Section 5.3, we repeat the experiments presented in Carmel et al. [2001b] by using pivoted TF-IDF. In Sections 5.4 and 5.5, we focus on a high-performing parameterized model (BM25) without and with parameter tuning. In Section 5.6 we repeat the experiments using a parameter-free model (DLHH). Section 5.7 summarizes the results obtained by setting the threshold ϵ to a default value (1). Finally, Section 5.8 tests our proposed pruning method on a bigger collection (.GOV), with a competitive model (BM25) on optimized settings, and using non-ad-hoc queries. The aim

is to see the effect of our technique on a different environment than ad hoc retrieval.

In the following sections, we present the results for the WT10G collection using MAP, in several figures. We provide figures for P@10 for pivoted TF-IDF results only, to keep the number of graphs low and because when a set of relevance judgements has already been created MAP is a more reliable effectiveness measure than P@10 [Sanderson and Zobel 2005]. In any case, those results with P@10 are reported to assess the good behavior of the pruning techniques with respect to that metric. Due to the high number of experiments and results, we present in an appendix the results for Disks 4 and 5, and results using an optimized setting for BM25 (Section 5.5) for the WT2G and LATimes collections, to avoid further graph overload. We report that those results confirm that the robustness of the PRP-based pruning technique is independent of the dataset employed.

5.2 Updating Document Lengths

The original formulation of the pruning algorithm described in Carmel et al. [2001b] is ruled by some theoretical guarantees. Starting from that method we define a baseline by improving its performance with a slight modification. The algorithm aims at keeping the search engine's top returned results, whether the original or the pruned inverted file is used. However, there are some restrictions that need to be fulfilled in order for these properties to hold. The score of the matching function should be the same before and after pruning in order to ensure the ranking-preserving theoretical guarantees. Pruning affects collection statistics (term document frequencies most notably) and document sizes (which should remain the same in order to preserve the aforementioned guarantees).

On the contrary, our belief is that collection and document statistics should be updated after pruning. Weighting functions are modelled under some assumptions over those statistics, which might be breaking up if the information contained in the index does not reflect the statistics in use. For instance, one document might be pruned much more than others (with many nonrelevant keywords), and not updating its document length would imply not retrieving it even if the remaining terms match some query. Next, we present a simple experiment comparing the performance of Carmel's pruning method when updating document lengths and when not updating them.

Figure 6 shows that updating the collection statistics in every pruning method brings better results, and thus a stronger baseline, for both short and long queries. Note that the top-k similarity preserving condition is therefore no longer guaranteed, but it performs better, with respect to MAP and P@10.

5.3 PRP-Based Pruning and Pivoted tf-idf Retrieval

The first batch of experiments mimicked the settings of the experiments reported in the first static pruning paper [Carmel et al. 2001b]. This implies using pivoted TF-IDF (Equation (17)) for retrieval, with *slope* set to 0.2. The pruning scores for Carmel's method were derived from pivoted TF-IDF as well (as implied

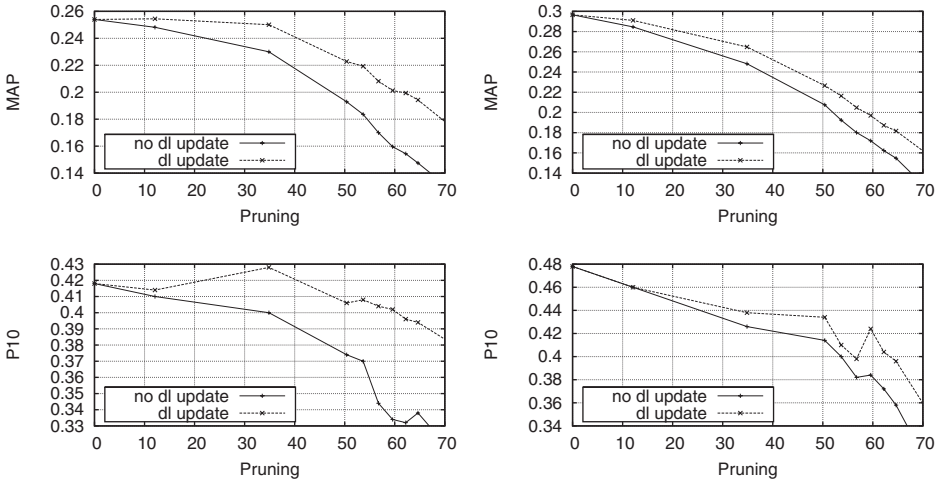

Fig. 6. Carmel method, WT2G, MAP, and P@10 for short(left) and long(right) queries BM25 retrieval updating and not-updating the document lengths.

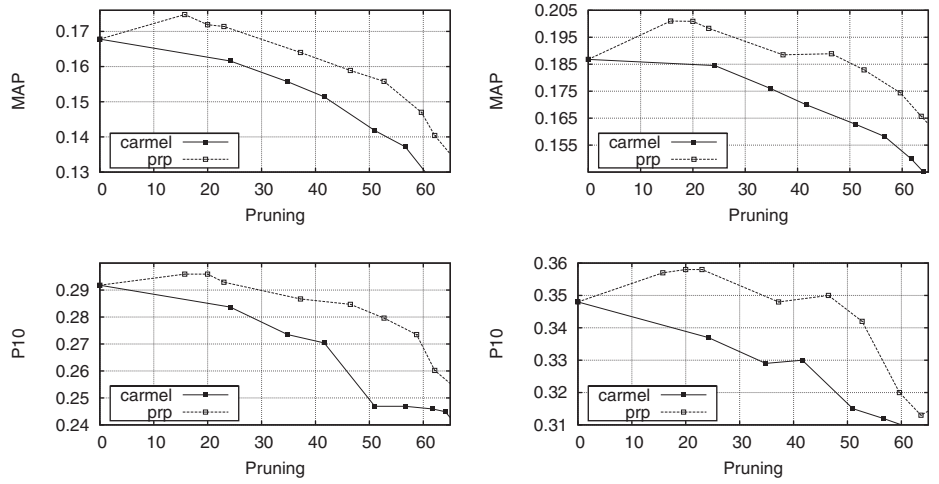

Fig. 7. PRP vs Carmel, short(left) and long (right) queries, WT10G, TF-IDF.

from the algorithm). As a consequence of the results presented in Section 5.2 our implementation updates the document lengths and the rest of the collection statistics, therefore implying a higher baseline for the comparison.

These results are presented in Figure 7. The PRP-pruning with this particular retrieval model outperforms the baseline for MAP and P@10 at all levels and for both query lengths, in the WT10G collection. Overall, the original precision values can be retained up to a 40–50% pruning level, and at earlier levels PRP-based pruning is able to improve them.

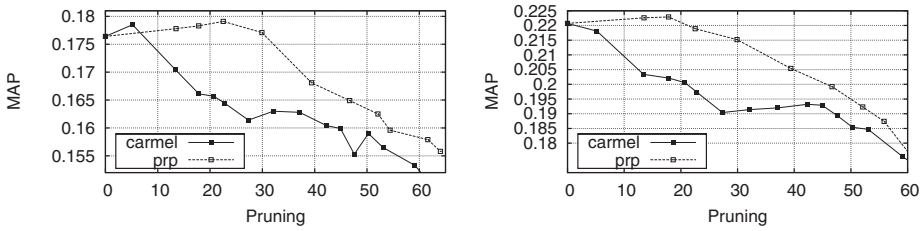

Fig. 8. PRP vs Carmel. BM25 retrieval with $b = 0.75$, short(left) and long (right) queries, WT10G.

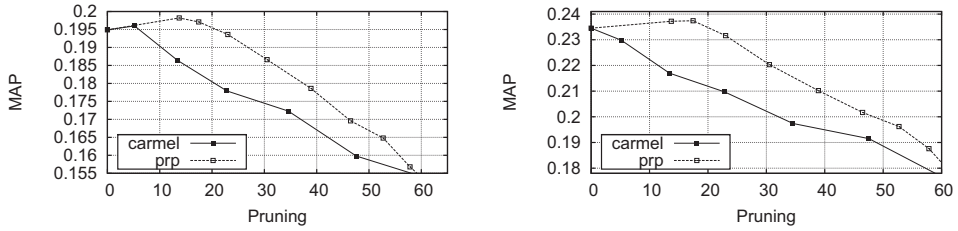

Fig. 9. PRP vs Carmel. BM25 retrieval with the best b, short(left) and long (right) queries, WT10G.

5.4 PRP-Based Pruning and BM25 Retrieval (Recommended Settings)

This second batch of experiments make use of BM25 [Robertson et al. 1995], a scoring function that performs better than pivoted TF-IDF in ordinary retrieval tasks, in most cases, and regardless of the metric employed for measuring performance. First, we set the b parameter to its recommended value ($b = 0.75$) to see how the pruning algorithms work in a default-settings scenario.

Figure 8 presents the result in the WT10G collection. Results are very consistent across collections. The PRP-based pruning performs as well as, or outperforms (in most cases the latter) the baseline. Overall, original precision can be increased for every collection, and the method behaves well for both short and long queries.

5.5 PRP-Based Pruning and BM25 Retrieval (Optimized Settings)

In order to rule out any possible bias introduced by the document length normalization effect, this batch of experiments optimizes the value of the b parameter in the BM25 formula, for every pruned index. This means that the MAP and P@10 results obtained are the best that the BM25 scoring function can achieve in every pruning level (for both the baseline and PRP-based pruning methods). The b value used for retrieval is optimized for both MAP and P@10. The b value Carmel's method uses for pruning was set to 0.75.

Results are shown in Figure 9. In this case, the probabilistic pruning outperforms the baseline in terms of MAP, and this effect is most likely to be independent of the normalization effect. Results are consistent for MAP (in most cases PRP-based pruning works better) across collections. With respect to P@10 our technique works clearly better for the WT10G collection; in other collections the behavior is not so clear: in disks 4 and 5 (Figure 20, Appendix A) the PRP-based method outperforms the baseline for short queries and long queries at

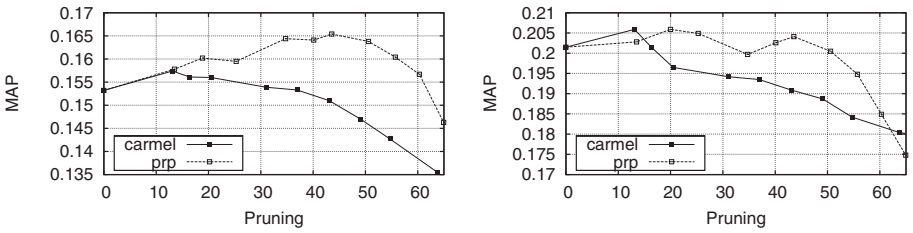

Fig. 10. PRP vs Carmel. DLHH retrieval, short(left) and long (right) queries, WT10G.

early pruning levels (<40%). In the WT2G collection (Figure 19, Appendix A) our technique works better than the baseline for short queries and worse for long queries. In the LATimes collection (Figure 18, Appendix A) it works better with short queries and high pruning levels.

5.6 Pruning Using a Parameter-Free Retrieval Model

In previous experiments we showed that our new pruning algorithm performs well with both TF-IDF and BM25, that these results are robust across different collections, and that the performance of our pruning technique is independent of a particular choice of parameters. In order to furthermore assess this last property, we focus our attention into a completely parameter-free retrieval model. BM25 and pivoted TF-IDF have parameters controlling the amount of normalization assigned to long documents. Even when we tuned those parameters for BM25 in Section 5.5, it is not totally clear whether pruning is discarding irrelevant (or noisy) data from the inverted file, or it is altering the collection statistics in a way that they might be more suitable for a particular weighting function. As well, it would be interesting to see how the pruning behaves when using a third matching function, as our pruning algorithm is score-independent (unlike the baseline). We employed DLHH [Amati 2006], which is a parameter-free matching function derived from the DFR framework [Amati and van Rijsbergen 2002].

Figure 10 presents the last batch of experiments. Results report that the parameter-free model benefits from pruning in every collection and with long and short queries. Hence, this experiment gives empirical support for the claim that pruning can bring beneficial parameter-independent effects in retrieval. It is particularly important to remark that the probabilistic pruning algorithm is able to improve the retrieval performance, even at high pruning levels, a fact that also occurred in many of the previous experiments.

5.7 Pruning with a Default Threshold

Table II summarizes the values obtained with the threshold set to $\epsilon = 1$ for the PRP-based pruning algorithm, as recommended in Section 3.1, in the WT10G collection. In this case, the pruning level resulted in $\approx 14\%$, and the resulting indexes produce gains in retrieval performance, despite the retrieval method employed, for both MAP and P@10. This supports the claim that controlled pruning can be beneficial for both efficiency and effectiveness of IR systems.

Table II. WT10G Collection. MAP and P@10 Values Obtained by Setting the Threshold ϵ to 1 Using PRP-Based Pruning, Compared to Those of an Unpruned Index. The Pruning Level is $\approx 14\%$

	WT10G							
	Short queries				Long queries			
	Original		Pruned		Original		Pruned	
	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
TF-IDF	0.1678	0.2918	0.1744	0.2939	0.1868	0.3480	0.1992	0.3560
BM25 $b = 0.75$	0.1764	0.2847	0.1779	0.2888	0.2207	0.3560	0.2231	0.3650
BM25 best b	0.1949	0.3112	0.1982	0.3122	0.2345	0.3650	0.2372	0.3720
DLHH	0.1532	0.2561	0.1581	0.2663	0.2015	0.3460	0.2032	0.3530

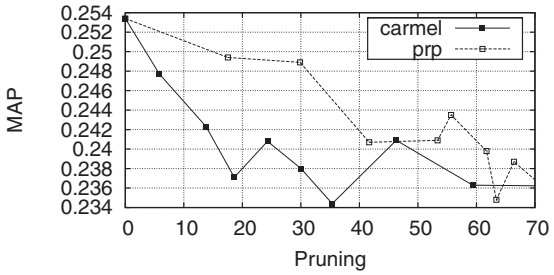

Fig. 11. .GOV collection, namepage-homepage finding + topic distillation. BM25 retrieval with the best b.

5.8 Experiments with the .GOV Collection

In this batch of experiments, we ran both pruning algorithms on the .GOV, a larger Web collection (actually WT10G has more documents, although they are shorter). The reason for these experiments is that we are performing retrieval with queries of a different nature, namely page finding and topic distillation. We did not employ any kind of query detection mechanism or link information to obtain the scores, just a raw BM25 (with the b parameter tuned like in Section 5.5). The topic set employed was the one developed for the TREC Web track 2004. It contains a mix between three types of queries: home page finding, named page finding, and topic distillation.

Figure 11 presents the results for both pruning regimes. The probabilistic pruning outperforms the baseline for most of the pruning levels, although this time the original MAP value could not be improved.

6. DISCUSSION

The final formulation for the PRP-based pruning presented in this article (Equation 7) could also be seen as a term-weighting function derived in a probabilistic fashion, much in the same way as language models. PRP pruning can therefore be stated as a uniform pruning method, using the same cut-off value for pruning every posting list. However, the benefits of this particular formulation are clear: if we run Carmel's method with either BM25 (pure probabilistic)

or TF-IDF, in every situation the PRP-pruned indexes behave better retrieval-wise (using the same scoring function for ranking). It is also important to stress that the pruning criterion (to decide which posting entries we keep/rule out) is stated in a completely different way than other pruning techniques.

The probabilities presented in Section 4 can be split into term-dependent and term-independent. The most discriminative factor in the derivation is the query likelihood probability. The effect of the document prior (term-independent probability), is to adjust the effect of the estimation for every document. For a given term and a fixed term frequency, the query-likelihood probability, $p(q_i|D,r)$ is penalized for longer documents (considering smoothing, it is a softened maximum-likelihood). When the $p(r|D)/(1-p(r|D))$ component is combined with the query likelihood, this value is softened for longer documents, so it is easier for the term to score below a certain threshold.

Term-dependent probabilities, $p(q_i|D,r)$ and $p(q_i|\overline{r})$, are the most influential elements for deciding whether or not the pruning algorithm is going to rule out a term-document entry from the index.

We are assuming that the estimations for the query likelihood and document prior are suitable for pruning purposes. This is supported by the fact that the smoothing method (linear interpolation) chosen for $p(q_i|D,r)$ is a well-known technique for LM-based retrieval, and the document prior is beneficial for retrieval.

The PRP-based pruning algorithm, as detailed in Section 4 involves a number of parameters: λ for the query likehood component (Equation (8)), average $\overline{X}_d$ and typical deviation S_d over the document lengths for the document prior (Equation (13)), and the result of the fit, Equation 16, (which uses term frequencies) and the number of documents N for the probability of a term given nonrelevance. Something remarkable is that, except for the parameter λ, all these parameters are derived from collection statistics. After some preliminary tests, we decided to empirically set λ to 0.6, for every experiment and every collection. Employing these settings implies that for all the runs there were no collection-dependent tuned parameters.

We performed an additional experiment in order to assess the robustness of the method with respect to the tunable parameters of scoring functions. We selected the WT10G collection and ran the PRP-based pruning algorithm with BM25, and for every pruned index we measured the performance using MAP for 10 different values of the b parameter. Then, we selected the median value for every pruning level and plotted both the median and best values in Figure 12. From the results obtained, it is clear that the median performance is close to the optimum, and therefore, we can conclude that the resulting indexes are robust for retrieval and not obtained just by performing collection-dependent parameter tuning. This statement is true for both the parameters involved in the pruning method and scoring function.

In Section 5 we compared our pruning method against a baseline of Carmel's pruning and also with respect to a full index (0% pruning). Our pruning method removed terms with $df > N/2$, however those stopwords were not removed from

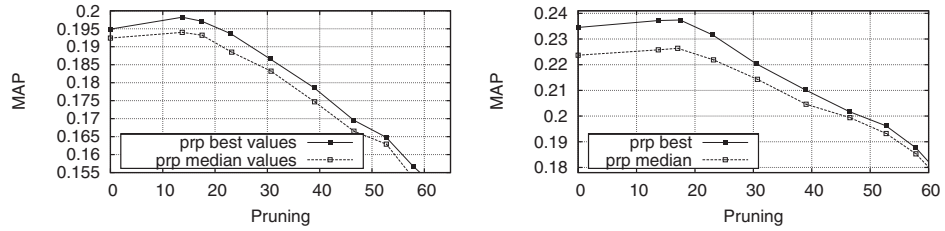

Fig. 12. BM25 retrieval median and optimal performance for all the b values employed in Section 5.5, short (left) and long (right) queries), WT10G collection.

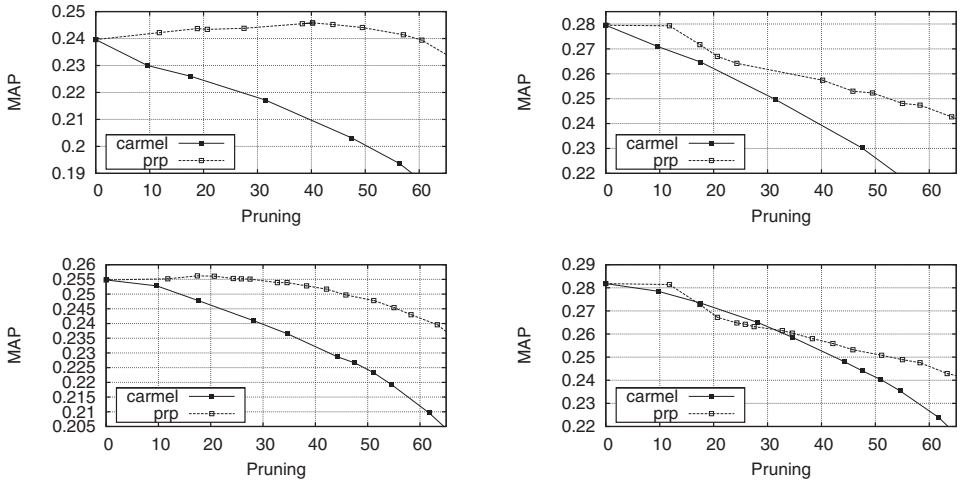

Fig. 13. PRP vs Carmel, short(left) and long (right) queries, Disks 4&5 BM25 with b = 0.75 (top) and with the best b (bottom).

the full index nor from Carmel's pruned index. Because PRP-based pruning is effectively automatically removing stopwords, it can be argued that its retrieval superiority over Carmel's method could be at least partially due to this fact. To investigate this point further, we performed an additional experiment with Disks 4 and 5 (Figure 13) and WT10G (Figure 14) collections. We comment next on the results for the TREC Disks 4 and 5 collection because it has the largest number of topics (250) and also the retrieval performance gains at early pruning levels (<40%) are very noticeable.

We compared both pruning methods by removing exactly the same terms (those with $df > N/2$) at an initial pruning level and ran both algorithms under those conditions. The first point on Carmel's method line in Figure 13 ($\approx 9\%$) is due just to the removal of those terms; the effect of pruning with $\epsilon = 1$ in PRP-based pruning (see Section 5.7) includes other posting entries removed in the index.

Removing those high-frequency terms improves Carmel's method performance and makes differences among both pruning techniques less noticeable. In any case PRP pruning still behaves better, specially with short queries.

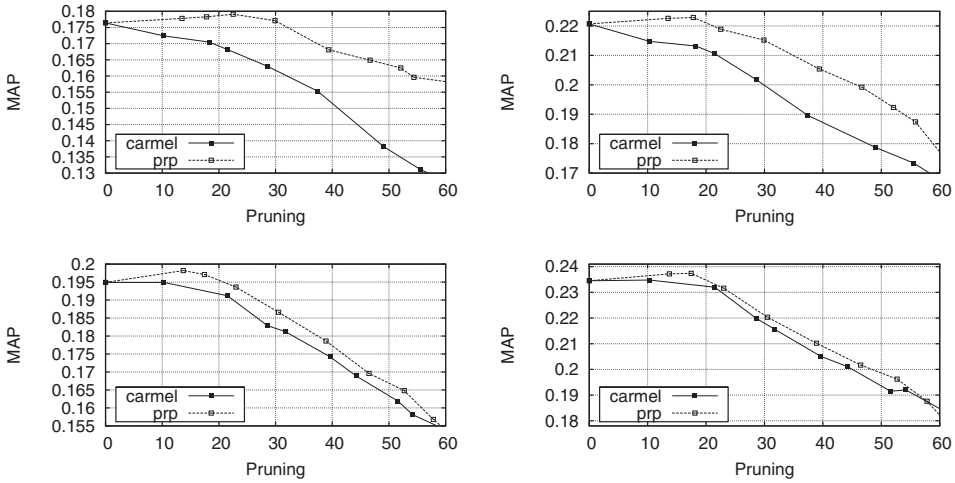

Fig. 14. PRP vs Carmel, short(left) and long (right) queries, WT10G BM25 with b = 0.75 (top) and with the best b (bottom).

We focus now on the number of removed *query* terms. We checked the number of pruned stopwords at every pruning level from the total number of query terms and total number of different query terms. Pruning and retrieving short queries using BM25 causes a noticeable improvement in retrieval precision at low pruning levels (see Figure 13). However, the proportion of the query terms' inverted lists pruned under the different methods is a very low percentage of all the query terms and it is constant through all the pruning levels. For pruning levels up to 80%, PRP-based pruning rules out 7 unique query terms out of 538 unique terms appearing in the 250 queries (less than 1.5%). If we consider all nonunique query terms, the number of removed terms is 24 out of 691 (less than 1.5%).

Any method that discards whole terms entries from the index has the risk of not being able to find any documents for a given query (the query "the who" for instance). If this is an issue, an elegant solution for keeping information on pruned terms could be using a two-tiered index [Ntoulas and Cho 2007]: a first layer normally pruned index and a secondary layer index that contains the pruned terms' information. The first index can be used for normal access and answering most of the queries whereas the secondary index would be accessed only if necessary.

Finally, we conducted an experiment in order to measure the similarity between the top 10 results with respect to the original index, varying the pruning levels. We employed a variation of Kendall's τ statistic [Kendall 1938] to compare the correlation of two ranked lists. This variation was presented by Fagin et al. [2003] and addresses the case of ranked lists that may contain different elements. This variation was also employed in the original evaluation of Carmel's method ([Carmel et al. 2001b]). Figure 15 shows the performance of both pruning methods in the WT10G collection, using short and long queries

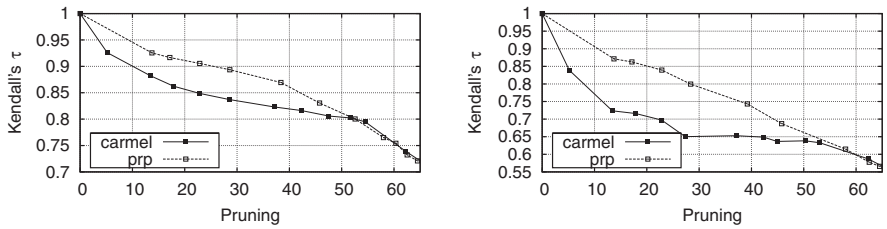

Fig. 15. Kendall's τ correlation on the top 10 results. PRP vs Carmel, short(left) and long (right) queries, WT10G , BM25.

and BM25 for retrieval. A τ measure of 1 implies a perfect one-to-one correlation between the top-k results, whereas 0 implies a total disagreement between the ranked lists.

The high similarity between the top 10 lists for moderate pruning levels supports the claim that for moderate pruning levels, the top 10 results of the pruned indices are very similar to the top 10 results of the original index. This holds both for PRP-based pruning and Carmel's method. It is remarkable that although the original design of PRP-based pruning was not guided by the principle of maintaining the top-k results, the final outcome of the method is even better than Carmel's.

7. CONCLUSIONS AND FUTURE WORK

The probability ranking principle is the foundation of the probabilistic model of IR, which retrieves documents relevant to a query (see Section 3.2). The fundamental idea behind this work is to introduce this principle for static pruning and not for the estimation of relevance. In this work we presented a novel index pruning algorithm based on the same assumptions as probabilistic retrieval models. Contrary to other pruning approaches, the technique presented here does not aim at maintaining the top results returned by an IR system. Instead, it defines a pruning criterion that relies on the goodness of the estimation of three probabilities.

In order to assess the performance of PRP-based pruning algorithm, we compared it to an enhanced version of Carmel et al.'s algorithm, using standard TREC datasets with queries of different lengths, and three different retrieval models with both default and optimized settings. The experiments showed that our technique outperforms the baseline in terms of MAP and performs at least as well in terms of P@10. Results are consistent across five different collections, 450 ad hoc queries of two different lengths, 225 topic distillation, and named page finding queries, and three retrieval models. Furthermore, a set of experiments allowed us to conclude that the outcome of the pruning algorithm does not depend on a particular parameter tuning of the retrieval model.

The fact that static pruning may improve retrieval precision up to certain pruning levels, may indicate a weakness of the scoring function. It is

possible to consider that a perfect matching function could leave the index as it is and prune it dynamically, but with high online query processing costs (this may be more critical for pruning regimes that operate in a document by document fashion). This can be critical in high-demand environments, or in systems deployed on low-memory devices. Furthermore, this could open a future line of work for retrieval modeling inspired by the results in pruning performance.

One of the weaknesses of the pruning method presented in this article is that it does not guarantee preserving the top-k results when using the original unpruned index. This is also the case for the original implementation of Carmel et al's pruning algorithm, which shifts the scores of posting entries and that was presented in Section 2, and also for the improved baseline presented in Section 5.2. However, if the estimation of the probabilities presented in Section 4 were ideal, then PRP-based pruning would bring the optimum amount of pruning for a given index. Therefore, as has also been stated experimentally, the precision values can be improved at moderate pruning levels. We further showed experimentally that PRP-based pruning behaves better than Carmel's method at maintaing the top-k results.

The approach proposed in this work follows a term-by-term mechanism, and it needs collection statistics that can only be obtained after indexing. For that reason, the pruning algorithm would operate in two different phases. This is a drawback with respect to the approach described in Büttcher and Clarke [2006] where this problem is solved by inspecting a subcollection and extrapolating the statistics found to the whole collection, and hence pruning can be done while indexing. This could be a future line of research. Forthcoming work could try to dig deeper into the nature and estimation of the probabilities or try to incorporate term cooccurrence into the pruning process in a formal way and also to assign different nonconstant costs per document in Equation (2). For instance, pruning reputably home-pages would imply higher costs than other kinds of documents.

An immediate future line of research is to design and perform suitable statistical tests for comparing different pruning methods. Note that traditional significance tests employed in retrieval pose a number of problems in this scenario. First of all, pruning algorithms do not produce a specific pruning level: both operate with a threshold, and the exact amount of pruning cannot be determined in advance. This makes difficult the comparison at the same pruning level between different methods or collections; it would be necessary to perform a trial-and-error procedure until both methods produce an index with exactly the same amount of pruning. This would allow for comparing single pruning levels. For obtaining a single value over all the pruning levels, a first step could be to interpolate the pruning level and compare the curves produced by the pruning methods on the basis of any variable. This could open the way for standard significance tests.

Finally, as stated in Section 6 it could be possible to alter the $p(q_i|\overline{r})$ estimation in such a way that it incorporates more refined stopword detection variants that could lead to even higher benefits from pruning.

APPENDIX A

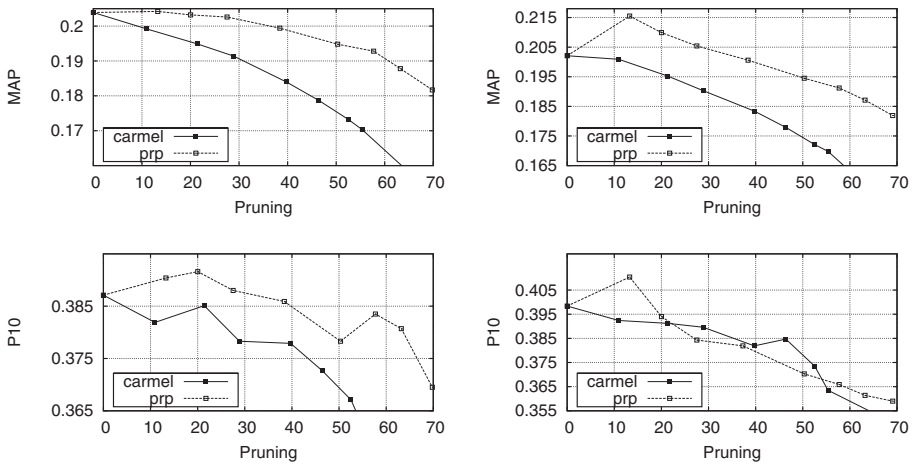

Fig. 16. PRP vs Carmel, short(left) and long (right) queries, Disks 4&5 , TF-IDF.

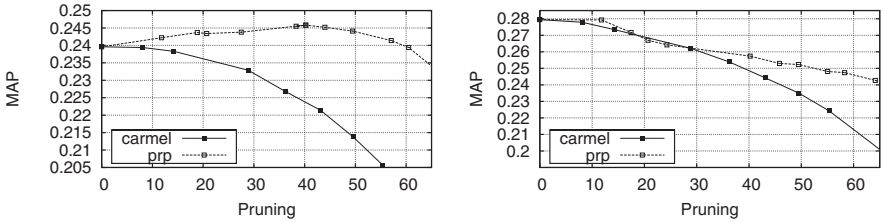

Fig. 17. PRP vs Carmel. BM25 retrieval with $b = 0.75$, short(left) and long (right) queries, Disks 4&5.

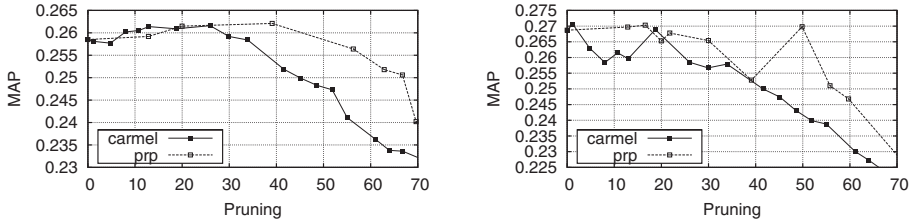

Fig. 18. PRP vs Carmel. BM25 retrieval with the best b, short(left) and long (right) queries, LATimes.

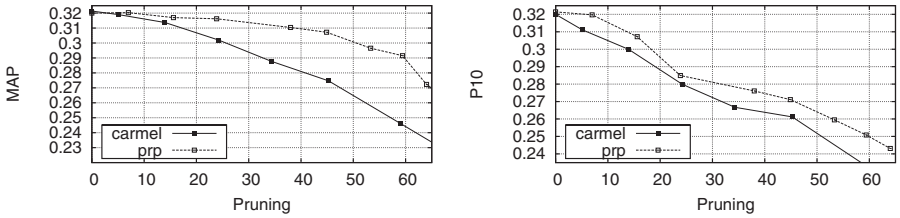

Fig. 19. PRP vs Carmel. BM25 retrieval with the best b, short(left) and long (right) queries, WT2G.

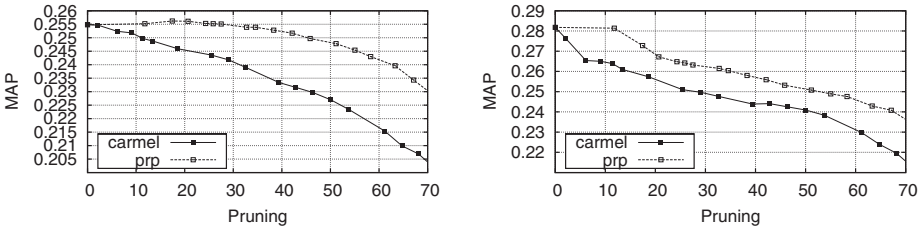

Fig. 20. PRP vs Carmel. BM25 retrieval with the best b, short(left) and long (right) queries, Disks4&5.

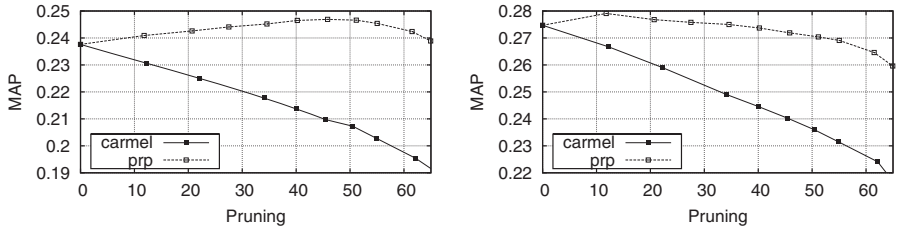

Fig. 21. PRP vs Carmel. DLHH retrieval, short(left) and long (right) queries, Disks4&5.

ACKNOWLEDGMENTS

We kindly thank David Losada and Christina Lioma for their help reviewing this article. We also wish to thank the three anonymous reviewers for their helpful suggestions on this article.

REFERENCES

AMATI, G. 2006. Frequentist and Bayesian approach to information retrieval. In *Proceedings of the 28th European Conference on IR Research (ECIR)*. Lecture Notes in Computer Science, vol. 3936. Springer, 13–24.

AMATI, G. AND VAN RIJSBERGEN, C. J. 2002. Probabilistic models of information retrieval based on measuring divergence from randomness. *ACM Trans. Inform. Syst. 20*, 4, 357–389.

ANH, V. N. AND MOFFAT, A. 2002. Impact transformation: effective and efficient Web retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 3–10.

ANH, V. N. AND MOFFAT, A. 2006. Pruned query evaluation using pre-computed impacts. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 372–379.

BAEZA-YATES, R., GIONIS, A., JUNQUEIRA, F., MURDOCK, V., PLACHOURAS, V., AND SILVESTRI, F. 2007. The impact of caching on search engines. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 183–190.

BAEZA-YATES, R. A. AND RIBEIRO-NETO, B. A. 1999. *Modern Information Retrieval*. ACM/Addison-Wesley.

BLANCO, R. AND BARREIRO, A. 2007a. Boosting static pruning of inverted files. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 777–778.

BLANCO, R. AND BARREIRO, A. 2007b. Static pruning of terms in inverted files. In *Proceedings of the 29th European Conference on IR Research (ECIR)*. Lecture Notes in Computer Science, vol. 4425. Springer, 64–75.

BRANDOW, R., MITZE, K., AND RAU, L. 1995. Automatic condensation of electronic publications by sentence selection. *Inform. Proc. Manag. 31*, 5, 675–685.

BRIN, S. AND PAGE, L. 1998. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the 7th International Conference on World Wide Web*. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 107–117.

BUCKLEY, C., SINGHAL, A., MITRA, M., AND SALTON, G. 1995. New retrieval approaches using smart: Trec. In *Proceedings of the 4th Text REtrieval Conference (TREC-4)*. 25–48.

BÜTTCHER, S. AND CLARKE, C. L. A. 2006. A document-centric approach to static index pruning in text retrieval systems. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, New York, NY, 182–189.

CARMEL, D., AMITAY, E., HERSCOVICI, M., MAAREK, Y. S., PETRUSCHKA, Y., AND SOFFER, A. 2001a. Juru at trec 10 - experiments with index pruning. In *Proceedings of the 10th Text REtrieval Conference (TREC)*. 228–236.

CARMEL, D., COHEN, D., FAGIN, R., FARCHI, E., HERSCOVICI, M., MAAREK, Y., AND SOFFER, A. 2001b. Static index pruning for information retrieval systems. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 43–50.

CHOWDHURY, A., MACCABE, M. C., GROSSMAN, D., AND FRIEDER, O. 2002. Document normalization revisited. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 381–382.

CRASWELL, N. AND HAWKING, D. 2004. Overview of the TREC-2004 Web Track. In *Proceedings of the 13th Text REtrieval Conference (TREC)*.

CROFT, W. AND HARPER, D. 1979. Using probabilistic models of document retrieval without relevance information. *J. Document. 35*, 4, 285–295.

FAGIN, R., KUMAR, R., AND SIVAKUMAR, D. 2003. Comparing top k lists. *SIAM J. Discr. Math. 17*, 1, 134–160.

HE, B. AND OUNIS, I. 2003. A study of parameter tuning for term frequency normalization. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*. ACM, New York, NY, 10–16.

KENDALL, M. 1938. A new measure of rank correlation. *Biometrica 30*, 81–89.

KRAAIJ, W., WESTERVELD, T., AND HIEMSTRA, D. 2002. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 27–34.

LAFFERTY, J. AND ZHAI, C. 2003. Probabilistic relevance models based on document and query generation. In *Language Modeling for Information Retrieval*, W. Croft and J. Lafferty Eds., Kluwer, 11–56.

LAVRENKO, V. AND CROFT, W. B. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 120–127.

LONG, X. AND SUEL, T. 2003. Optimized query execution in large search engines with global page ordering. In *Proceedings of the 29th International Conference on Very Large Data Bases*. VLDB Endowment, 129–140.

MANMATHA, R., RATH, T., AND FENG, F. 2001. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 267–275.

NTOULAS, A. AND CHO, J. 2007. Pruning policies for two-tiered inverted index with correctness guarantee. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 191–198.

ROBERTSON, S. 1977. The probability ranking principle in IR. *J. Document. 33*, 294–304.

ROBERTSON, S. AND SPARCK-JONES, K. 1976. Relevance weighting of search terms. *J. Amer. Soc. Inform. Sci. 27*, 129–146.

ROBERTSON, S., VAN RIJSBERGEN, C. J., AND PORTER, M. F. 1981. Probabilistic models of indexing and searching. In *Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval (SIGIR)*. Butterworth & Co., Kent, UK, 35–56.

ROBERTSON, S. AND WALKER, S. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval)*. ACM, New York, NY, 232–241.

ROBERTSON, S. E., WALKER, S., HANCOCK-BEAULIEU, M., GULL, A., AND LAU, M. 1995. Okapi at TREC-4. In *Proceedings of the 4th Text REtrieval Conference (TREC-4)*. 73–86.

SAKAI, T. AND SPARCK-JONES, K. 2001. Generic summaries for indexing in information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 190–198.

SALTON, G., WONG, A., AND YANG, C. S. 1975. A vector space model for automatic indexing. *Commu. ACM 18*, 11, 613–620.

SANDERSON, M. AND ZOBEL, J. 2005. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 162–169.

SINGHAL, A., BUCKLEY, C., AND MITRA, M. 1996. Pivoted document length normalization. In *Proceedings of the 19st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 21–29.

STROHMAN, T., TURTLE, H., AND CROFT, W. B. 2005. Optimization strategies for complex queries. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 219–225.

THEOBALD, M., WEIKUM, G., AND SCHENKEL, R. 2004. Top-k query evaluation with probabilistic guarantees. In *Proceedings of the 13th International Conference on Very Large Data Bases*. VLDB Endowment, 648–659.

TURTLE, H. AND FLOOD, J. 1995. Query evaluation: strategies and optimizations. *Inform. Proc. Manag. 31*, 6, 831–850.

UPSTILL, T., CRASWELL, N., AND HAWKING, D. 2003. Query-independent evidence in home page finding. *ACM Trans. Inform. Syst. 21*, 3, 286–313.

VAN RIJSBERGEN, C. J. 1977. A theoretical basis for the use of co-occurrence data in information retrieval. *J. Document. 33*, 106–119.

VAN RIJSBERGEN, C. J. 1979. *Information Retrieval*. Butterworths, London.

VOORHEES, E. M. AND HARMAN, D. K. 2005. *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press.

WESTERVELD, T., KRAAIJ, W., AND HIEMSTRA, D. 2002. Retrieving Web pages using content, links, URLS and anchors. In *Proceedings of the 10th Text Retrieval Conference (TREC-10)*. 663–672.

WITTEN, I. H., MOFFAT, A., AND BELL, T. C. 1999. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, San Francisco, CA.

ZHAI, C. AND LAFFERTY, J. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inform. Syst. 22*, 2, 179–214.

ZOBEL, J. AND MOFFAT, A. 2006. Inverted files for text search engines. *ACM Comput. Surv. 38*, 1–56.

Received January 2008; revised July 2008, October 2008; accepted November 2008