Upper-Bound Approximations for Dynamic Pruning

CRAIG MACDONALD and IADH OUNIS, University of Glasgow, UK
NICOLA TONELLOTTO, Information Science and Technologies Institute, National Research Council of Italy (ISTI-CNR)

Dynamic pruning strategies for information retrieval systems can increase querying efficiency without decreasing effectiveness by using upper bounds to safely omit scoring documents that are unlikely to make the final retrieved set. Often, such upper bounds are pre-calculated at indexing time for a given weighting model. However, this precludes changing, adapting or training the weighting model without recalculating the upper bounds. Instead, upper bounds should be approximated at querying time from various statistics of each term to allow on-the-fly adaptation of the applied retrieval strategy. This article, by using uniform notation, formulates the problem of determining a term upper-bound given a weighting model and discusses the limitations of existing approximations. Moreover, we propose an upper-bound approximation using a constrained nonlinear maximization problem. We prove that our proposed upper-bound approximation does not impact the retrieval effectiveness of several modern weighting models from various different families. We also show the applicability of the approximation for the Markov Random Field proximity model. Finally, we empirically examine how the accuracy of the upper-bound approximation impacts the number of postings scored and the resulting efficiency in the context of several large Web test collections.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Performance, Experimentation

Additional Key Words and Phrases: Dynamic pruning, upper bounds

ACM Reference Format:
Macdonald, C., Ounis, I., and Tonellotto, N. 2011. Upper-bound approximations for dynamic pruning. ACM Trans. Inf. Syst. 29, 4, Article 17 (November 2011), 28 pages.
DOI = 10.1145/2037661.2037662 http://doi.acm.org/10.1145/2037661.2037662

1. INTRODUCTION

Web search engines allow billions of documents to be searched, with near-instantaneous response time. To achieve this, they must exploit efficient retrieval techniques to minimize the time and resources required to score documents for queries. Web searchers often look only at the top few pages of results for a query [Silverstein et al. 1998]. For this reason, the complete scoring of every document that contains at least one query term causes unnecessary latency, because not all of these documents will make the top retrieved set of documents that the user will view.

N. Tonellotto acknowledges the partial support of S-CUBE (EU-FP7-215483), ASSETS (CIP-ICT-PSP-250527), and VISITO Tuscany (POR-FESR-63748) projects, as well as an international travel grant from the Royal Society (TG090399).
Authors' addresses: C. Macdonald and I. Ounis, Department of Computing Science, University of Glasgow, Lilybank Gardens, G12 8QQ, Glasgow, Scotland, U.K.; emails: {craig.macdonald, iadh.ounis}@glasgow.ac.uk; N. Tonellotto, Information Science and Technologies Institute, National Research Council of Italy, Via G. Moruzzi 1, 56124 Pisa, Italy; email: nicola.tonellotto@isti.cnr.it.

DOI 10.1145/2037661.2037662 http://doi.acm.org/10.1145/2037661.2037662

Query-pruning strategies can increase efficiency by removing the low-scoring documents in the early stages of the query-scoring process. Some pruning strategies are *safe-up-to-rank-K* [Turtle and Flood 1995], meaning that the ranking of documents up to rank K will have full possible effectiveness, but with increased efficiency. In contrast, a strategy that can incorrectly rank documents before rank K is only approximate. As information retrieval (IR) is primarily concerned with effectiveness, in this work, we consider only safe-up-to-rank-K pruning strategies.

Pruning strategies can be implemented statically by altering the index structure at index construction time [Blanco 2008], or dynamically, at query scoring time. These safe-up-to-rank-K dynamic pruning strategies rely on maintaining a threshold score that documents must overcome in order to be considered in the top-K documents. To exploit this threshold, these strategies require that each term is associated with an upper bound on the maximal contribution of the weighting model to any document's relevance score. If the query terms present in a document do not have a cumulative upper bound higher than the current threshold, the document can be safely ignored.

Two methods for computing term upper bounds exist: first, the exact least upper bound may be calculated for all terms at indexing time using a particular weighting model [Broder et al. 2003; Croft et al. 2009]; second, the upper bound for each query term may be approximated as accurately as possible at retrieval time using various statistics of the term, without the need for precomputation.

In most of the previous research and application of efficient retrieval, all the returned search results of queries are ranked using a single weighting model. In such scenarios, the term upper bounds can be precomputed at index-building time, tying the IR system to a single or several preselected weighting models. However, recently there has been a trend towards selective approaches to information retrieval in which the exact retrieval approach taken for each query varies. For instance, recent works [Kang and Kim 2003; Geng et al. 2008] showed that it was beneficial to exploit different ranking models for different queries, calling this process *query-dependent ranking*. The context of a search query often provides a search engine with meaningful hints to better answer the current query [Xiang et al. 2010]. Indeed, the task of *query segmentation* [Bendersky et al. 2009] results in different weighting schemes for a term depending on the context represented by the remaining query terms. Then the parameters of the weighting model, or the weighting-model itself, may require changing [Li et al. 2009] to allow per-query alterations of the weighting model setting, as per [He and Ounis 2004] or online adaptation of its parameters [Taylor et al. 2006]. In such selective retrieval approaches, the precalculation of upper bounds at indexing time may not be sufficiently agile. Instead, approximations of the term upper bounds at retrieval time represent a valid alternative to indexing time precomputation of the upper bounds.

In this work, we study the accurate approximation of the upper bounds of weighting models suitable for the MaxScore and Wand dynamic pruning strategies. The accuracy of the upper-bound approximations are important, as they can impact the effectiveness or the efficiency of query-scoring strategies: too high, and some documents will be unnecessarily scored (reduced efficiency); too low, and some documents will not be fully scored (unsafe, reduced effectiveness). While the calculation of least upper bounds may have been trivial in the past for simple TF.IDF weighting models, their accurate approximation for modern, non-trivial weighting models is unclear. In particular, one approximation involving a parameter was proposed for weighting models based on each term's IDF [Broder et al. 2003]. However, this approximation is inapplicable for weighting models such as those from language modeling [Zhai and Lafferty 2004] and Divergence From Randomness (DFR) [Amati 2006], which cannot be suitably factored into document-variant and document-invariant parts.

Hence, it is clear that the approximation of upper bounds for other weighting models is an open problem. In this paper, we propose a new upper-bound approximation, and prove that is safe for several recent weighting models by the application of a constrained nonlinear maximization problem. This upper-bound approximation can be calculated on-the-fly at query execution time. By adopting the proposed safe upper-bound approximation, search engines can experience enhanced efficiency without loss of effectiveness. The contributions of this article are as follows: we review existing upper bounds using a uniform notation; we show how the upper-bound approximation problem can be modeled as a constrained nonlinear maximization problem; we study the problem with three representative weighting models and derive an upper-bound approximation that is proven to be safe; we examine the extent to which the upper-bound approximations approach a ground truth of the actual least upper-bound, by measuring their efficiency for several dynamic pruning strategies using many queries on two different large-scale TREC test collections; lastly, we investigate the approximation of upper bounds for a state-of-the-art term proximity model.

In the remainder of this article, Section 2 reviews dynamic pruning strategies. Section 3 reviews existing approximations for upper bounds using a uniform notation. We tackle the upper-bound approximation problem for three weighting models using constrained nonlinear maximization in Section 4. Section 5 contains experiments on the efficiency of the approximations. In Section 6, we examine the appropriateness of upper-bounding proximity weighting models. We provide concluding remarks in Section 7.

2. DYNAMIC PRUNING

In most IR systems, the relevance score for a document d given a query Q follows the general outline given by the best match strategy:

$$\text{score}_Q(d, Q) = \omega\, S(d) + \kappa \sum_{t \in Q} \text{score}(tf_d, *_d, *_t), \tag{1}$$

where $S(d)$ is the combination of some query-independent features of the document d (e.g. PageRank, URL length), and $\text{score}(tf_d, *_d, *_t)$ is the application of a weighting model to score tf_d occurrences of term t in document d. $*_d$ denotes any other document statistics required by a particular weighting model, such as document length. Orthogonally, $*_t$ represents any statistics of the term necessary for the weighting model, such as the number of documents in which the term occurs, so that IDF can be calculated. More than one static feature may be applied to compute the static score $S(d)$, with learning-to-rank techniques [Liu 2009] used to find appropriate weights for the features (e.g. ω, κ). In this article, we focus on the effective and efficient calculation of $\text{score}(tf_d, *_d, *_t)$, yet our experiments also consider when static features are and are not present in the final ranking function.

The scoring of a document as per Equation (1) requires processing the postings lists of each query term in the inverted index. The algorithms to match and score documents for a query fall into two main categories [Moffat and Zobel 1996]: in *term-at-a-time (TAAT)* scoring, the query term posting lists are processed and scored in sequence, so that documents containing term t_i gain a partial score before scoring commences on term t_{i+1}. In contrast, in *document-at-a-time (DAAT)* scoring, the query term postings lists are processed in parallel, such that all postings of document d_j are considered before scoring commences on d_{j+1}. Compared to TAAT, DAAT has a smaller memory footprint than TAAT due to the lack of maintaining intermediate scores for many documents, and is reportedly applied by large search engines [Anagnostopoulos et al. 2005].

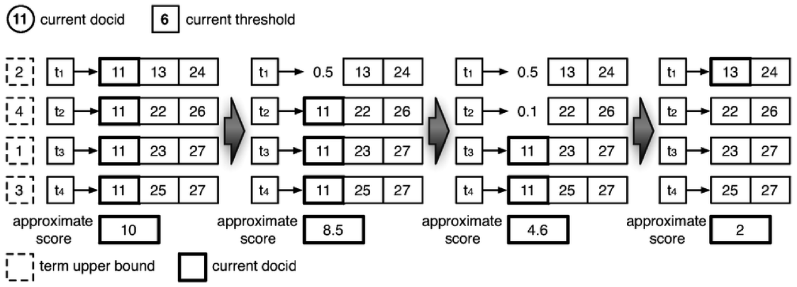

Fig. 1. How the DAAT MaxScore strategy processes a document.

An alternative strategy to DAAT and TAAT is *score-at-a-time* [Anh and Moffat 2001]; however, this is suitable only for indices sorted or partially sorted by document importance, which must be calculated before the actual query processing. Hence, this strategy shares the limitations of other static-pruning strategies; for example, it precludes on-the-fly adaption to support different ranking models or parameters.

To increase the efficiency of full-scoring TAAT or DAAT, two dynamic pruning strategies for TAAT and DAAT which are safe-up-to-rank-K were proposed by Turtle and Flood [1995], which increase efficiency by early termination of document scoring. In DAAT MaxScore, the scoring of a document is omitted if it is possible to guarantee that the document will never obtain a score greater than the minimum score of the current top-K documents. The algorithm keeps track of the K-th largest document score observed as a threshold that candidate documents must exceed before they can enter the partial ranking. Moreover, the actual scoring of each document is optimized, as explained with the help of Figure 1.

In Figure 1, the terms $t_1 \cdots t_4$ are ordered by decreasing document frequency. Each term has an upper bound on the maximum score that any document containing the given query term can obtain. When a document is processed, its approximate score is initially assumed to be the sum of the approximate scores of terms appearing in the same document. Then, the approximate score is updated each time a posting is processed, using the exact score instead of the approximate one. As soon as the approximate score falls below the current threshold, the current document is guaranteed that it can not make it into the top K, so no more postings with the same document require scoring and the algorithm can move on to the next document. Analogously, in TAAT MaxScore, no new terms are scored for retrieval once it is possible to guarantee that any new document can never obtain a score greater than the minimum score of the current top-K partially scored documents. Both strategies are safe-up-to-rank-K, meaning that retrieval effectiveness cannot be impaired by the application of these strategies, while markedly improving efficiency [Turtle and Flood 1995]. An improved version of TAAT MaxScore has been proposed by Persin [1994], where two thresholds rule the insertion of new partial scores in the top-K results and their partial score update. Although this strategy shows good efficiency improvements, the heuristics used to determine the thresholds lead to potentially unsafe-up-to-rank-K results.

The choice of query semantics deployed by the search engines can impact both retrieval safeness and efficiency. In particular, in conjunctive processing (where all query terms must exist in a retrieved document), recall can be negatively impacted. Indeed, a relevant document may not be retrieved even though it contains all but the least important query term. However, conjunctive processing allows pruning to take place more aggressively, as documents that do not contain a query term can be discarded immediately [Moura et al. 2008; Skobeltsyn et al. 2008; Altingovde et al. 2009]. In this

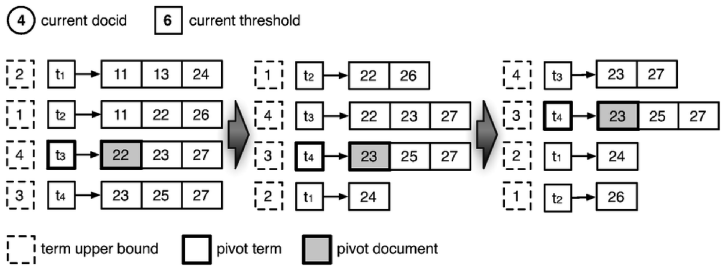

Fig. 2. How the WAND strategy selects the next document to score.

article, we focus exclusively on disjunctive retrieval, as recently it has been shown not to produce significantly different high-precision effectiveness compared to conjunctive retrieval [Craswell et al. 2011]. Nevertheless, the work in the following sections on term upper bounds is equally applicable to conjunctive processing. We intend to experimentally compare and contrast the effect of term upper bounds for conjunctive and disjunctive processing in future work.

DAAT WAND [Broder et al. 2003]—the most recent safe-up-to-rank-K dynamic pruning strategy—is only suitable for disjunctive query processing, but works by determining how close to conjunctive processing can be obtained for a given query without loss of effectiveness. Similarly to MAXSCORE, WAND maintains the current top-K documents and a threshold equal to their minimum score. For any new document, WAND calculates an approximate score, summing up the upper bounds for the terms occurring in the document. If this approximate score is greater than the current threshold, then the document is fully scored. It is then inserted in the top-K candidate document set if this score is greater than the current threshold, and the current threshold is updated. If the approximate score check fails, the next document is processed. The selection of the next document to score is optimized and explained with the help of Figure 2.

In Figure 2, the set of posting lists for the terms $t_1 \cdots t_4$ are maintained in increasing order of the docid that each posting list currently refers to. Then a *pivot term* is computed, that is, the first term for which the accumulated sum of upper bounds of preceding terms and itself exceeds the current threshold (e.g., term t_3 with accumulated score of 7). The corresponding docid in the posting list of the pivot term identifies the *pivot document*, that is, the smallest docid having a chance to overcome the current threshold. If the current docids of the previous terms are equal to the pivot document docid, the document will be fully scored. Otherwise, the posting list of one of the preceding terms is moved to the pivot document docid, and the procedure is repeated. In the example, a good candidate document is found at the third step (23) and is fully processed. In contrast to DAAT MAXSCORE, DAAT WAND can benefit from skipping every posting list to a particular document to reduce disk IO.

Algorithms similar to dynamic pruning strategies have also been proposed in the database research community—such as Fagin's algorithm, the threshold algorithm, and the no random-access algorithm [Fagin et al. 2003]. While based on the same thresholding mechanism, these assume that the posting lists are sorted in descending order of score (in a similar manner to static pruning), and in some cases require random access to the posting list. In contrast, IR systems access postings lists in a sequential manner to minimise disk seek overheads [Moffat and Zobel 1996].

TAAT MAXSCORE, DAAT MAXSCORE, and DAAT WAND all rely on maintaining a threshold score at query scoring time that documents must overcome to be considered in the top-K documents. To guarantee that an early termination by the dynamic pruning strategy will provide the correct top-K documents, it is necessary to calculate, for each

term, an upper bound on its maximal contribution to the score of any document in its posting list. In the following section, we use a uniform notation for expressing upper bounds and their approximations, and discuss the limitations of existing approximations, in particular with respect to their appropriateness for modern weighting models.

3. ANALYSIS OF EXISTING UPPER BOUNDS

In this section, we describe a uniform notation for defining the upper bounds for a term, formulate existing methods for obtaining upper bounds within this notation, and discuss their limitations. In particular, an upper bound for term t is denoted $\sigma(t)$. Within this section, we review the two main methods for obtaining upper bounds: the least upper bound for a given weighting model can be empirically determined at indexing time (Section 3.1); alternatively, an upper bound can be approximated at querying time using appropriate statistics (Section 3.2).

3.1. Least Upper Bounds

The least upper bound (that would be observed for all occurrences of a term) for all terms may be obtained at indexing time, given prior knowledge of the weighting model and a scan of the posting list of each term [Broder et al. 2003]. In particular, the least upper bound $\sigma_{\mathsf{LEAST}}(t)$ for term t, is obtained using all documents in the posting list $L(t)$

$$\sigma_{\mathsf{LEAST}}(t) = \max_{d \in L(t)} \text{score}(tf_d, *_d, *_t), \tag{2}$$

where $\text{score}(tf_d, *_d, *_t)$ is the score given by a weighting model for tf_d occurrences of term t in document d. $*_d$ denotes any other document statistics required by a particular weighting model, such as document length (which we denote l_d). $*_t$ represents any statistics of the term necessary for the weighting model, such as IDF.

However, the traversal of an entire index to determine the least upper bound for each term has some disadvantages: first, such pre-computation may add overhead to the indexing phase, as the entire inverted index must be traversed and scores recorded for every weighting model likely to be used during retrieval.[1]

Moreover, the precomputation means that the pre-specified settings of the weighting model(s) cannot be altered. However, increasingly, selective retrieval approaches are being devised which apply different rankings for different queries. For example, using one ranking model for all the queries cannot be a suitable solution to conquering the search ranking challenge [Zhu et al. 2009]. As was also found by Kang and Kim [2003], a good ranking algorithm for an informational does not always perform well for a homepage finding task. Inspired by these observations, if queries could be divided properly into different groups, it is possible to design different ranking models for each group to improve the search ranking. A more recent work [Geng et al. 2008] showed that it was beneficial to exploit different ranking models for different queries, calling this process *query-dependent ranking*, and in He and Ounis [2004] weighting models are adapted on-the-fly for different queries. Moreover, the context of a search query often provides a search engine with meaningful hints for better answering the current query [Xiang et al. 2010]. Indeed, the task of *query segmentation* [Bendersky et al. 2009] results in different weighting schemes for a term depending on the context represented by the remaining query terms, and the weighting models can be trained online in light of new relevance data [Croft et al. 2009].

[1]For instance, prescoring an index of the 50 million English documents from the TREC ClueWeb09 collection to obtain least upper bounds takes 40 min on a single machine.

If all weighting models and their parameter settings[2] are identified in advance, then the precomputation of upper bounds for all models is feasible. However, instead of the precomputation of upper bounds, this work centers on the alternative of approximating the upper bounds at querying time, allowing the IR system to be more agile and flexible.

3.2. Upper-Bound Approximations

This work concerns the approximation (e.g., A, $\sigma_{\mathsf{A}}(t)$) of upper bounds, using statistics of t and some knowledge of the applied weighting model, such that $\sigma_{\mathsf{LEAST}}(t) \leq \sigma_{\mathsf{A}}(t)$. Such statistics can be easily computed at indexing time, without prior knowledge of the weighting model to be applied at retrieval time.

First, we note that Fang et al. [2004] describe various heuristics that are common to effective weighting models, relating to the effects of varying tf_d, l_d, and term statistics. These heuristics include several that are of interest in obtaining upper-bound approximations.

TFC1. Favour a document with more occurrences of a query term than one with less (assuming equal document lengths).

TFC2. Ensure that the change in the score caused by increasing tf_d from 1 to 2 is larger than that caused by increasing tf_d from 100 to 101 (assuming equal document lengths).

LNC1. Penalize long documents (assuming equal tf_d).

In the following, we describe how various upper bounds can be approximated, referring back to these heuristics when appropriate. The accurate approximation of these upper bounds is critical to the efficiency and effectiveness of the TAAT MaxScore, DAAT MaxScore, and DAAT Wand strategies. In particular, using term upper bounds, the strategy can know when it is not worth scoring a given posting for a particular term, as even the maximal contribution that the term could give would not impact the final top-K ranked documents. However, if a term upper bound is too high, then some postings will be needlessly scored, when they would not have any impact on the top ranking of K results. If a term upper bound is too low, then some postings may incorrectly be omitted from scoring, potentially impacting on the ranking of the top-K results—making the method only approximate instead of safe-up-to-rank-K. Conversely, $\sigma_{\mathsf{A}}(t) = \infty$ would be safe, but very inefficient, as it would prevent any pruning from taking place. In approximating the upper bounds, we desire the smallest $\sigma_{\mathsf{A}}(t) \geq \sigma_{\mathsf{LEAST}}(t)$, to ensure safeness, but maximize efficiency.

Term upper bounds were first proposed by Turtle and Flood [1995], in relation to the MaxScore approaches. However, the authors did not discuss the calculation of these upper bounds. We assume that the calculations of such upper bounds were determined as trivial in the presence of simple TF.IDF models, which did not account for the length of documents (i.e., LNC1 was not considered). In particular, for TF.IDF, score$(tf_d, *_d, *_t)$ is monotonically increasing as $tf_d \to \infty$, as specified by TFC1. Hence the least upper bound is easily found using the maximum term frequency of term t in its posting list $L(t)$ ($\max_{d \in L(t)} tf_d$):

$$\sigma_{\mathsf{LEAST:TFIDF}}(t) = \text{score}\left(\max_{d \in L(t)} tf_d, *_d, *_t\right), \tag{3}$$

where tf_d is the term frequency of term t in document d, and $*_d$ is empty (i.e. document length is not required).

[2]We note that for many weighting models, the effect of varying its parameters will have an unquantifiable effect on the least upper bound values for terms. Hence, least upper bounds must be calculated for each applied parameter setting.

However, modern weighting models often account for document length (l_d), in the manner of LNC1. In this case, the approximation of the term upper bounds needs to account for document length, and/or infer upper bounds using knowledge of the weighting model. Due to the interaction of TFC1 and LNC1, it is hard to analytically determine $\sigma_{\mathsf{LEAST}}(t)$ for such weighting models. Instead, for instance, in Broder et al. [2003], the authors of WAND observed that score$(tf_d, *_d, *_t)$ can be factored into a product $w(tf_d, *_d) \cdot w(*_t)$. While the authors do not specify the weighting model that they use, they state that $w(*_t)$ reflects the IDF component of the weighting model. We note that TF.IDF and BM25 [Robertson et al. 1992] can be factored in this manner, with $w(tf_d, *_d)$ reflecting the factor which varies with d. Then, an approximate upper bound can be found:

$$\sigma_{\mathsf{Factor}}(t) = C \cdot w(t), \tag{4}$$

for some tunable constant C. We call this approximation Factor. The experiments described in Broder et al. [2003] did not tune C, as it depends on the weighting model applied. Instead, a factor on the current threshold score (which has an inverse relation to C) was varied, demonstrating for the WT10G collection the efficiency/effectiveness tradeoffs for aggressive but approximate pruning. Hence, the proper setting of C to achieve a safe supremum of $\sigma(t)$ was not investigated. It is of note that Broder et al. [2003] did not discuss the use of $\max_{d \in L(t)} tf_d$ in calculating approximate upper bounds. For example, the upper bounds of BM25 (see Equation (7)) can be approximated by $\sigma_{\mathsf{Factor:BM25}}(t) = C \cdot (k_1 + 1) \cdot \frac{(k_3+1)qtf}{k_3+qtf} \log_2 \frac{N-N_t+0.5}{N_t+0.5}$.

Later, in Lacour et al. [2008], the upper bounds were approximated for the BM11 weighting model which does consider document length l_d, by replacing l_d with the average length of all documents avg_l:

$$\sigma_{\mathsf{AVGDL}}(t) = \text{score}(\max_{d \in L(t)} tf_d, avg\_l, *_t). \tag{5}$$

In doing so, this approximation (AVGDL) assumes a uniform document length for every posting. However, in their experiments, the retrieval performance of safe-up-to-rank K techniques were impacted at rank K, inferring that strategies such as TAAT MAXSCORE and DAAT MAXSCORE were being impacted by inexact approximation of the term upper bounds, causing the strategies to become approximate strategies only, and negatively impacting effectiveness.

All of the existing methods to find upper bounds have limitations: $\sigma_{\mathsf{TFIDF}}(t)$ makes no consideration of document length; Factor is limited to weighting models which can be suitably factored—in particular, it is inapplicable to weighting models such as those from language modelling (LM) [Zhai and Lafferty 2004] and Divergence From Randomness (DFR) [Amati 2006]; in contrast, AVGDL may be inexact and not safe. In the next section, we model the term upper-bound approximation problem and perform a mathematical proof of the safeness of upper-bounds for several state-of-the-art weighting models.

4. NEW UPPER-BOUND APPROXIMATIONS

Modern weighting models, such as BM25 and those from the language modeling and DFR families, take into account document length (l_d) in addition to the term frequency (tf_d), either as part of a document length normalization process (LNC1), or as part of maximum likelihood estimation ($\frac{tf_d}{l_d}$). In these cases, $\sigma_{\mathsf{LEAST}}(t)$ cannot be exactly calculated using the $\max_{d \in L(t)} tf_d$ alone without any knowledge of the weighting model.

To counteract the limitations of the Factor and AVGDL approximations, we formulate, in general terms, the problem of approximating a term upper bound given the weighting model. We study the problem for three weighting models from three different families

and propose a new upper bound suitable for these weighting models. In particular, we focus on: BM25 [Robertson et al. 1992]—an example of a TF.IDF-based weighting model with document length normalization; Dirichlet language modeling (LM) [Zhai and Lafferty 2004]—which smooths the maximum likelihood ($\frac{tf_d}{l_d}$) using the probability of occurrence in the collection; and the parameter-free DLH13 [Amati 2006] from the DFR family of weighting models—which also includes document length normalization.

To simplify the notation, in the following we use x to denote tf_d, y to denote l_d and Greek letters (α, β, γ) to denote strictly positive constants.

Given a general weighting model $f(x, y)$ depending on term frequency (x) and document length (y), the problem of finding an upper bound for $f(x, y)$ can be formulated as a *constrained maximization problem* (CMP) [Jongen et al. 2004]:

$$\max f(x, y), \text{ subject to } \begin{cases} x \leq y, \\ x_{min} \leq x \leq x_{max}, \\ y_{min} \leq y \leq y_{max}, \\ x, y \in \mathbb{N}, \end{cases} \tag{6}$$

where $x_{min} > 0$, $y_{min} > 0$, $x_{max} > x_{min}$, $y_{max} > y_{min}$, $x_{max} > y_{min}$, and $y_{max} > x_{max}$ are reasonable assumptions for any real world IR system and document corpus. For instance, an IR system based on an inverted index will not score documents where a query term does not occur (hence $x_{min} > 0$), and similarly, empty documents (i.e. $y_{min} = 0$) will not be scored. Moreover, for any given document, any term can appear a number of times not greater than the total number of tokens in the document $(x \leq y)$. Finally, the upper bound on the number of occurrences on a given term in any document, and the length of any document in a posting list are finite $(x \leq x_{max}, y \leq y_{max})$ and countable. Without loss of generality, we assume $x_{min} = y_{min} = 1$. These constraints define the admissible region of the problem, that is, the area of the x, y plane where the term statistics have acceptable values.

Our approach for approximating upper bounds uses statistics for each term (e.g., x_{max}, y_{max}), and uses these to calculate the score of a special document which would define an upper bound on the score of any permissible posting for that term. These statistics are easily calculable at indexing time, and not specific to any particular weighting model or setting. Then, based on the CMP defined by the constraints in Equation (6), we analyze to determine the position of the special document in the x, y plane which has maximal score. Such a special document probably does not exist, and hence the least upper bound will lie at some other point inside the admissible region; nevertheless, the approximate upper bound is guaranteed to be safe.

The CMP defined above is integral and nonlinear, and hence very difficult to manipulate. Since we are interested in an approximate upper bound, we relax the problem by removing the $x, y \in \mathbb{N}$ constraint. In this case, the admissible region defined by the remaining constraints is regular because the constraints are affine [Jongen et al. 2004]. Hence, the problem can be studied using constrained optimization methods. In the following, we study the analytical behaviour of the BM25, LM, and DLH13 weighting models on the relaxed admissible region, and analytically derive the solutions of the relaxed constrained maximization problems by using first order differential calculus.

In BM25 [Robertson et al. 1992], the relevance score of a document d for a query term t is given by

$$\text{score}(x, y, N_t) = \frac{(k_1 + 1)x}{k_1((1 - b) + b\frac{y}{avg\_l}) + x} \frac{(k_3 + 1)tf_q}{k_3 + tf_q} w^{(1)}, \tag{7}$$

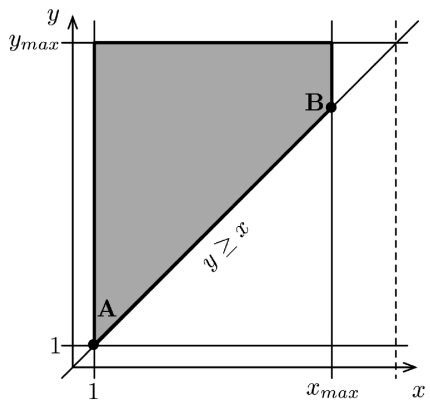

Fig. 3. Admissible region for the relaxed CMP.

where tf_q is the frequency of the query term t in the query; b, k_1, and k_3 are parameters (defaults $b = 0.75$, $k_1 = 1.2$, and $k_3 = 1000$ [Robertson et al. 1992]). $w^{(1)}$ is the IDF factor, which is given by $w^{(1)} = \log_2 \frac{N - N_t + 0.5}{N_t + 0.5}$. N is the number of documents in the whole collection. N_t is the document frequency of term t.

THEOREM 1. *The solution of the relaxed CMP for Equation* (7) *is given by* $x = x_{max}$ *and* $y = x_{max}$.

PROOF. Equation (7) can be conveniently rewritten as

$$f(x, y) = \beta \frac{(k_1 + 1)x}{k_1\big((1 - b) + b\frac{y}{\alpha}\big) + x} = \frac{\beta' x}{x + \alpha' y + \gamma},$$

where $\alpha' = \frac{k_1 b}{avgl}$, $\beta' = (k_1 + 1)\frac{(k_3+1)tf_q}{k_3+tf_q} w^{(1)}$, $\gamma = k_1(1 - b)$. It can be seen that this function is strictly monotonically decreasing in y, as suggested by LNC1, and strictly monotonically increasing in x, as per TFC1.

Figure 3 is a graphical representation of the constraints in Equation (6). The shaded region identifies the admissible region where the maximum point must lie, bounded by $1 \le x \le x_{max}$, $1 \le y \le y_{max}$, and $y \ge x$. The segment of $y = x$ within these bounds is denoted *AB*. Next, because $f(x, y)$ is monotonically decreasing in y and strictly monotonically increasing in x, it follows that the maximum of Equation (7) is reached on the segment *AB*, because for any other point in the admissible region, it is always possible to find another point closer to *AB* with a greater value of $f(x, y)$. Along the segment *AB* (where $y = x$), we have

$$f(x, y)\Big|_{y=x} = f(x) = \frac{\beta' x}{(1 + \alpha')x + \gamma},$$
$$\frac{\partial f}{\partial x}\Big|_{y=x} = \frac{df}{dx} = \frac{\beta' \gamma}{((1 + \alpha')x + \gamma)^2}.$$

Hence, for $y = x$, $f(x)$ has a positive, continuous derivative, meaning that its maximum is reached at B, where $x = x_{max}$ and $y = x_{max}$. □

In Dirichlet language modeling [Zhai and Lafferty 2004], the maximum likelihood of a term t occurring in the document model is smoothed to the collection model. Applying a log transformation to convert the product of probabilities into a summative best

match model results in

$$\text{score}(x, y, F_t) = tf_q \cdot \log\left((1 - \lambda_{LM})\frac{x}{y} + \lambda_{LM}\frac{F_t}{T_C}\right), \tag{8}$$

where T_C is the number of tokens in the collection, F_t is the frequency of the query term in the collection, and $\lambda_{LM} = \frac{\mu}{\mu+y}$ is the form of the Dirichlet smoothing (where μ is a positive parameter).

THEOREM 2. *The solution of the relaxed CMP for Equation* (8) *is given by* $x = x_{max}$ *and* $y = x_{max}$.

PROOF. Equation (8) can be conveniently rewritten as

$$f(x, y) = tf_q \cdot \log\left(\frac{x + \beta\mu}{y + \mu}\right),$$

where $\beta = \frac{F_t}{T_C} < 1$. This function is strictly monotonic increasing in x and strictly monotonic decreasing in y because they appear only in the numerator and denominator of the logarithm function, respectively. Then, according to the same discussion as for Theorem 1, its maximum lies on the AB segment from Figure 3. Then we have

$$f(x, y)\Big|_{y=x} = f(x) = tf_q \cdot \log\left(\frac{x + \beta\mu}{x + \mu}\right),$$
$$\frac{\partial f}{\partial x}\Big|_{y=x} = \frac{df}{dx} = \frac{(1 - \beta)\mu}{x^2 + (\beta + 1)\mu x + \beta\mu^2}.$$

Next, as $\mu > 0$ [Zhai and Lafferty 2004] and $\beta < 1$, the derivative is always positive for $x > 0$. Then, $f(x)$ is increasing along the segment AB and is maximal for $x = x_{max}$ and $y = x_{max}$. □

The DLH13 weighting model is a generalization of the parameter-free hypergeometric DFR model in a binomial case [Amati 2006], given as

$$\text{score}(x, y, F_t) = \frac{tf_q}{x + 0.5}\left(x \log_2\left(\frac{x \cdot N \cdot avg\_l}{y \cdot F_t}\right) + \frac{1}{2}\log_2\left(2\pi x\left(1 - \frac{x}{y}\right)\right)\right), \tag{9}$$

where avg_l is the average document length, calculated over the whole collection. Notably, DLH13 is parameter-free.

THEOREM 3. *The solution of the relaxed CMP for Equation* (9) *is given by* $x = x_{max}$ *and* $y = x_{max} + \frac{1}{2}$.

PROOF. With the change of variable $z = \frac{x}{y}$, Equation (9) can be conveniently rewritten as

$$f(x, z) = \frac{tf_q}{x + 0.5}\left(x \log_2(\beta z) + \frac{1}{2}\log_2(2\pi x(1 - z))\right),$$

where $\beta = \frac{N \cdot avg_l}{F_t}$. Deriving $f(x, z)$ w.r.t z, we obtain

$$\frac{\partial f}{\partial z} = \frac{tf_q}{\ln 2}\frac{(2x + 1)z - 2x}{z(2x + 1)(z - 1)}.$$

This derivative is 0 when $z = \frac{2x}{2x+1}$ corresponds to the line $y = x + \frac{1}{2}$. This line is just 0.5 units above the constraint $y = x$, and it lies inside the admissible region. It is easy to verify that, for a fixed $x = x^*$, the point $y^* = x^* + \frac{1}{2}$ is a maximum point for the function $f(x^*, y)$. Following the BM25 analysis, we can now study the function:

$$f(x,y)\Big|_{y=x+\frac{1}{2}} = f(x) = \frac{tf_q}{x+0.5}\left(x\log_2\left(\beta\frac{2x}{2x+1}\right) + \frac{1}{2}\log_2\left(\pi\frac{2x}{2x+1}\right)\right).$$

It is easy to see that this function is monotonically increasing, hence the maximum is reached when $x = x_{max}$, with the corresponding $y = x_{max} + \frac{1}{2}$. □

Note that the upper bound-approximation for DLH13 is almost identical to that found for BM25 and LM. The $\frac{1}{2}$ quantity takes into account the fact that DLH13 is undefined on the line $y = x$. Hence, we can summarize the results by proposing the following approximation:

$$\sigma_{\mathsf{MAXTF}}(t) = \text{score}(x_{max}, x_{max} + \tau, t), \tag{10}$$

where $\tau = 0$ for BM and LM and $\tau = \frac{1}{2}$ for DLH13. (Note that while $\tau = \frac{1}{2}$ is mathematically founded for DLH13, in general, $\tau > 0$ may be suitable for other weighting models that are undefined when $x = y$.)

The approximate upper bounds identified by the previous analysis are 'optimal' in the admissible region, that is, the largest value possible subject to the constraints of the relaxed CMP. However, in general they are unlikely to coincide with the least upper bounds calculated explicitly over all occurrences of a term in a posting list. Moreover, these bounds are 'pessimistic'—for example, in a two-term query, we are assuming the existence of a document entirely composed by the first term and a document entirely composed by the second term. However, should these documents exist, it is feasible that the documents would reach the top-K documents, if the pruning threshold is low enough.

Nevertheless, the application of these upper bounds permits these weighting models to be applied in a safe manner, because no other occurrence of a term can have a score larger than the proposed upper bound. It is of note that the only statistic required to be stored at indexing time is the maximal term frequency x_{max}, which is easily calculable at indexing time without prior knowledge of the weighting model or any parameters. The extent to which the upper bounds overestimate the least upper bound $\sigma_{\mathsf{LEAST}}(t)$ will depict the number of postings that will be extraneously scored, and impact efficiency. The extent that this occurs will be empirically investigated in the next section, using thorough experiments on two standard TREC test collections. Success can be interpreted by the efficiency of dynamic pruning strategies applied, measured in terms of postings scored and in average query latency.

In general, our results may be extended to other weighting models with the following properties: the score is monotonically increasing with respect to term frequency (TFC1) and monotonically decreasing with respect to document length (LNC1). For example, the analysis outlined in Theorem 2 can be carried out for a LM model using Jelinek-Mercer smoothing, obtaining identical results. However, this might not be suitable for some weighting models; for example, models which hinder cannot easily be derived in closed form (e.g., PL2 from DFR, which uses approximations for the binomial function). Nevertheless, for such models, it may be possible to use a more sophisticated approach,

Table I. Applied TREC Web Test Collections

Collection	# Documents	# Terms	Queries
GOV2	25,205,179	15,466,363	TREC 2005-2007 Terabyte
CW09B	50,220,423	74,238,222	TREC 2009 Web

such as using second order conditions [Jongen et al. 2004]—which we leave to future work.

5. APPROXIMATIONS EVALUATION

In the following, we test the accuracy and efficiency of the approximation presented in Section 4 when applied to the three dynamic pruning strategies that we test (namely TAAT MaxScore, DAAT MaxScore, and DAAT Wand). In particular, the accuracy of the approximations will affect the number of postings scored by the various dynamic pruning strategies, and hence impact on their overall efficiency. In this section we address the following research questions.

(1) What is the numerical accuracy of the approximations, measured in terms of correctness of the upper-bound approximation and the average overestimation; that is, by how much does the approximation exceed the actual upper bounds $\sigma_{\mathsf{LEAST}}(t)$?
(2) For different dynamic pruning strategies and safe-to-rank-K values, how many extra postings does each upper-bound approximation cause to be scored, compared to the actual upper bounds $\sigma_{\mathsf{LEAST}}(t)$?
(3) How efficient are the dynamic pruning strategies when using these upper bounds compared to the actual upper bounds $\sigma_{\mathsf{LEAST}}(t)$?
(4) How does the introduction of static scores (such as PageRank) into the retrieval process impact the upper-bound approximations?

Section 5.1 defines the experimental setup. Section 5.2 addresses the first research question by examining the numerical accuracy of the approximations. Section 5.3 examines how the accuracy of the approximations impacts the number of postings actually scored by the various dynamic pruning strategies (research question 2). Section 5.4 examines the resulting efficiency in terms of average query response time (research question 3). Finally, in Section 5.5 we address research question 4 by examining the impact of adding static scores to the retrieval process.

5.1. Experimental Setting

Experiments are performed using the three weighting models (BM25, LM, and DLH13) and two large-scale TREC Web test collections, namely GOV2 and the 50 million English document subset of the TREC ClueWeb09 (CW09B) [Clarke et al. 2010]. Their statistics are given in Table I. In our experiments, we use the Terrier IR platform.[3] Both GOV2 and CW09B corpora are indexed, applying Porter's English stemmer and removing standard stopwords. Positional information is not stored, and each posting consists of only the Elias-Gamma encoded document id gap and the Elias-Unary encoded frequency.

Both TAAT MaxScore and DAAT Wand can take advantage of the presence of additional skip pointers [Moffat and Zobel 1996] in the inverted file. These permit an advancement in the posting list to a given document, instead of only to the next posting. In doing so, the postings for documents that will never be retrieved will not be decompressed, thus increasing efficiency. Hence, we add a single level of skipping pointers [Moffat and Zobel 1996] to the inverted index (skipping parameter $L = 10{,}000$ for GOV2, $L = 100{,}000$ for CW09B, set empirically to maximize efficiency).

[3]http://terrier.org.

Table II. Test Queries Distribution per Number of Terms

Queries	Total	# of Query Terms						
		1	2	3	4	5	6	7
TREC Terabyte 2005	994	30	210	322	240	134	41	17
MSN 2006	965	209	360	244	98	39	15	—

During retrieval, the parameters for BM25 and LM (k_1, k_3, b, μ) remain at their default settings—in contrast, DLH13 is parameter-free. Efficiency experiments are made using a quad-core Intel Xeon 2.6GHz, with 8GB RAM, and a 250GB SATA disk holding the index.

For both GOV2 and CW09B corpora, we experiment both with and without static scores within the retrieval process. In particular, we use PageRank scores for each document, normalized to maximum 1, and weighted by parameter $\omega = 2$ (found by empirically maximizing mean average precision on 50 queries with relevance assessments of the TREC 2009 Web track). For the integration of the static document scores into the dynamic pruning strategies, TAAT MaxScore and DAAT Wand both require the use of an upper bound on the maximum value of the static score (i.e. $\omega = 2$). As alluded to in research question (4) above, this increases uncertainty about whether or not a document or term posting list can be pruned and hence impact the number of postings scored.

The queries used during retrieval depends on the test collection. For GOV2, we used the first 1000 queries from the TREC Terabyte track efficiency task [Buttcher et al. 2007]. We removed empty queries, queries with no results returned, and the six queries with more than seven terms (which we considered not to be significant) to obtain a final set of 994 queries. For CW09B, we extract a stream of user queries from a real search engine log. In particular, we select the first 1000 queries of the MSN 2006 query log [Craswell et al. 2009]. We removed empty queries, queries with no results returned, and the fifteen queries with more than six terms (which we again considered not to be significant), to obtain a final set of 965 queries. In all of the following experiments, results will be broken down by the number of query terms. The distributions of queries per number of terms are summarized in Table II.

5.2. Approximation Accuracy

The numerical accuracy of the upper-bound approximation will have an impact on the safeness and the efficiency of dynamic pruning strategies. If an approximation is lower than the least upper bound, then some postings may incorrectly be omitted from scoring, potentially making the ranking only approximate instead of safe-up-to-rank-K. If an approximation is greater than the least upper bound, the ranking is safe-up-to-rank-K, but if the value is far greater than the least upper bound value, then some postings will be needlessly scored.

We measured the least and approximated upper bounds for every term in both query logs, and for each approximation we report in Table III how many upper-bound approximations were not lower than the least upper bound and, for safe upper-bound approximations, the average relative increase introduced by the approximation. As the DLH13 and LM models cannot be factored into document variant and document invariant factors, we omit results for the Factor approximation.

From the results, we note that the approximation Factor performs well for BM25, with no unsafe upper bounds. This is expected, as it was explicitly designed to be used in conjunction with BM25.

Next, approximation AVGDL is unsafe in several cases, but the average relative increase of the approximation value is not very marked, with a notable exception for DLH13. However, this approximation can be unsafe for terms which have maximal term

Table III. Accuracy of the Term Upper-Bound Approximations, in Terms of Safeness and Average Relative Increases

	BM25			LM			DLH13		
Approx.	# Safe Terms	# Unsafe Terms	Avg. Rel. Increase	# Safe Terms	# Unsafe Terms	Avg. Rel. Increase	# Safe Terms	# Unsafe Terms	Avg. Rel. Increase
GOV2									
Factor	1926	0	128.5%	—	—	—	—	—	—
AVGDL	1885	41	2.2%	1901	25	23.3%	1869	57	*
MAXTF	1926	0	3.1%	1926	0	20.6%	1926	0	20.8%
CW09B									
Factor	1231	0	151.4%	—	—	—	—	—	—
AVGDL	1142	89	5.3%	1163	68	12.2%	1131	100	*
MAXTF	1231	0	10.2%	1231	0	8.8%	1231	0	15.6%

*Denotes that the value diverges to positive infinity due to the formulation of DLH13, while Factor is unsuitable for models other than BM25.

frequency less than avg_l. For instance, 'apotheosis' is a valid query term, but occurs at most 114 times in a CW09B document. The least upper bound value for BM25 is 12.49, but as $avg\_l = 698$, AVGDL unsafely approximates 12.44. This emphasises the fact that AVGDL is not suitable for ensuring safe-to-rank-K retrieval.

Overall, approximation MAXTF is the only safe approximation, as expected. In general, term upper-bounds (approximated or least) are calculated independently for each term, as the dynamic pruning strategy cannot know the maximum score of any document where the two terms co-occur. Thus, the upper bound will always be higher than necessary and on the safe side. For our proposed term upper-bound approximation, the average upper-bound approximation is always no higher than 21% above the least upper bound. However, it is not clear how this difference will impact the number of postings scored or the resulting efficiency. These two points are investigated in detail in the following two sections.

5.3. Approximation Impact on Posting Pruning

The accuracy of the upper-bound approximations will have an impact on the effectiveness and efficiency of the dynamic pruning strategies by affecting which postings are scored. We measure this, to address the second research question outlined above. In particular, we first precompute the least upper bound for each term. Then, for each dynamic pruning strategy, we measure the number of postings that were scored for each query term in any full strategy (i.e., TAAT FULL and DAAT FULL). Then we measure the percentage of postings that were scored for each query term using the least upper bounds with respect to the full strategies as a ground truth. Next, we test the MAXTF approximation and measure the percentage of postings scored. Compared to the least upper bounds, the percentage of total postings scored by the approximation should ideally be as close to those scored when using the least upper bound, such that time is not wasted scoring useless postings that should have been pruned.

Tables IV and V detail the percentage of postings scored using the LEAST and MAXTF upper bounds for each collection and dynamic pruning strategy, for $K = 1000$ and $K = 20$, respectively. Similarly, Figures 4 and 5 report the percentage of total postings extraneously scored by the MAXTF approximation compared to the least upper bound for each dynamic pruning strategy, again for $K = 1000$ and $K = 20$, respectively. Finally, to aid analysis, Table VI summarizes the mean percentage of postings scored by the least upper bound, and the increase when using the MAXTF approximation, across each experimental dimension (K, corpus, weighting model, dynamic pruning strategy, and query length) in Tables IV and V, while varying the other dimensions.

From the results, we make several observations. The number of postings scored by BM25 are low for GOV2 and CW09B using TAAT MAXSCORE (less than 38%). In this

Table IV. Percentage of Postings Scored by the LEAST and MAXTF Upper-Bound Approximations, Broken Down by Query Length ($K = 1000$)

Model	Strategy	Approx.	# of Query Terms					
			2	3	4	5	6	7
GOV2								
BM25	TAAT	LEAST	32.40%	29.42%	28.91%	29.74%	33.72%	30.65%
	MaxScore	MAXTF	32.72%	29.42%	28.91%	29.81%	34.10%	30.65%
	DAAT	LEAST	27.22%	11.95%	6.68%	5.08%	4.67%	3.33%
	MaxScore	MAXTF	27.29%	12.04%	6.73%	5.13%	4.72%	3.37%
	DAAT	LEAST	23.07%	9.22%	6.87%	6.47%	6.30%	4.78%
	Wand	MAXTF	23.19%	9.35%	6.98%	6.71%	6.42%	4.95%
LM	TAAT	LEAST	65.44%	49.75%	57.76%	63.61%	64.33%	64.94%
	MaxScore	MAXTF	95.44%	71.51%	62.15%	67.47%	71.96%	75.28%
	DAAT	LEAST	57.89%	33.25%	27.45%	26.29%	25.94%	29.48%
	MaxScore	MAXTF	87.11%	62.28%	43.44%	35.51%	34.39%	39.86%
	DAAT	LEAST	58.62%	38.34%	39.91%	41.03%	41.59%	49.66%
	Wand	MAXTF	87.17%	64.94%	52.17%	49.78%	51.05%	62.10%
DLH13	TAAT	LEAST	60.48%	46.21%	56.04%	59.25%	59.59%	63.13%
	MaxScore	MAXTF	75.49%	52.81%	58.24%	65.98%	65.23%	64.64%
	DAAT	LEAST	41.04%	25.49%	21.29%	19.03%	20.39%	19.48%
	MaxScore	MAXTF	59.79%	34.57%	27.82%	26.41%	25.91%	27.81%
	DAAT	LEAST	36.83%	30.64%	31.37%	29.21%	33.20%	34.35%
	Wand	MAXTF	58.95%	38.62%	39.04%	40.25%	40.78%	45.23%
CW09B								
BM25	TAAT	LEAST	32.99%	37.27%	19.80%	25.91%	25.15%	—
	MaxScore	MAXTF	32.99%	37.35%	19.80%	25.91%	25.15%	—
	DAAT	LEAST	13.95%	17.91%	5.86%	7.36%	4.42%	—
	MaxScore	MAXTF	13.96%	17.92%	5.87%	7.37%	4.43%	—
	DAAT	LEAST	15.02%	29.00%	8.00%	17.94%	12.41%	—
	Wand	MAXTF	15.04%	29.02%	8.04%	17.96%	12.43%	—
LM	TAAT	LEAST	50.32%	69.33%	61.25%	80.67%	82.23%	—
	MaxScore	MAXTF	60.69%	77.42%	63.97%	82.02%	86.43%	—
	DAAT	LEAST	49.78%	43.15%	35.22%	49.13%	46.79%	—
	MaxScore	MAXTF	58.86%	50.94%	43.46%	54.93%	51.32%	—
	DAAT	LEAST	52.13%	61.02%	50.05%	71.99%	74.04%	—
	Wand	MAXTF	60.80%	67.56%	56.95%	74.78%	76.76%	—
DLH13	TAAT	LEAST	44.97%	67.09%	60.86%	78.38%	81.18%	—
	MaxScore	MAXTF	49.66%	68.68%	61.86%	82.02%	81.18%	—
	DAAT	LEAST	38.10%	29.71%	26.00%	37.17%	33.19%	—
	MaxScore	MAXTF	43.38%	35.40%	29.37%	44.27%	40.52%	—
	DAAT	LEAST	39.03%	46.97%	41.04%	60.51%	59.04%	—
	Wand	MAXTF	45.12%	53.84%	44.70%	67.69%	66.77%	—

case, there is no marked difference between the least and approximate upper bounds, due to the pruning decision made by TAAT MaxScore. In particular, when it decides to prune a whole posting list comparing the current threshold with the term upper bound, a small difference on that value is unlikely to change the pruning decision. For the DAAT dynamic pruning strategies, the number of postings scored are also sensibly reduced with both upper bounds for BM25. For GOV2, the number of postings scored decreases for larger queries; however, the same effect is not present with real Web queries used with CW09B. Nevertheless, the MAXTF approximation gives very good results with respect to the least upper bounds for BM25. The percentage of scored postings exceeds the value in the least upper bound case by less than 0.6% for GOV2 ($K = 20$) and than 0.15% for CW09B ($K = 20$). If we limit to DAAT strategies, these bounds lower to 0.24% and 0.04%, respectively. The value of K does not have any noticable impact in this case. Overall, with mean increases in postings scored of 0.08% above the least upper bound (see Table VI), it is clear that the Factor approximation is entirely suitable for BM25.

Table V. Percentage of Postings Scored by the LEAST and MAXTF Upper-Bound Approximations, Broken Down by Query Length ($K = 20$)

Model	Strategy	Approx.	# of Query Terms					
			2	3	4	5	6	7
			GOV2					
BM25	TAAT	LEAST	27.93%	25.84%	22.73%	25.10%	29.25%	24.95%
	MaxScore	MAXTF	28.42%	26.44%	23.32%	25.10%	29.45%	24.95%
	DAAT	LEAST	20.09%	8.85%	4.48%	3.08%	2.73%	1.91%
	MaxScore	MAXTF	20.12%	8.86%	4.51%	3.09%	2.80%	1.93%
	DAAT	LEAST	15.64%	4.02%	2.07%	1.56%	1.95%	1.48%
	Wand	MAXTF	15.66%	4.07%	2.11%	1.61%	2.30%	1.53%
LM	TAAT	LEAST	37.16%	43.57%	50.01%	44.84%	47.91%	50.57%
	MaxScore	MAXTF	61.55%	45.55%	55.98%	58.27%	58.89%	58.54%
	DAAT	LEAST	21.63%	16.97%	12.95%	11.11%	10.82%	10.71%
	MaxScore	MAXTF	39.11%	21.83%	20.46%	19.66%	18.84%	19.04%
	DAAT	LEAST	18.07%	22.99%	20.84%	19.71%	20.36%	23.76%
	Wand	MAXTF	38.70%	28.80%	33.08%	33.19%	32.99%	36.72%
DLH13	TAAT	LEAST	34.63%	43.71%	45.54%	42.56%	49.54%	52.71%
	MaxScore	MAXTF	54.93%	43.94%	53.40%	49.86%	54.58%	57.24%
	DAAT	LEAST	21.01%	14.41%	9.60%	8.30%	8.97%	8.88%
	MaxScore	MAXTF	24.87%	18.70%	14.69%	11.84%	12.78%	13.63%
	DAAT	LEAST	16.15%	17.27%	13.73%	14.33%	16.37%	18.48%
	Wand	MAXTF	19.47%	25.42%	23.19%	20.19%	23.83%	28.26%
			CW09B					
BM25	TAAT	LEAST	23.13%	22.62%	12.92%	16.31%	7.65%	—
	MaxScore	MAXTF	23.14%	22.77%	12.92%	16.31%	7.65%	—
	DAAT	LEAST	12.13%	11.03%	3.24%	3.11%	1.81%	—
	MaxScore	MAXTF	12.13%	11.05%	3.26%	3.11%	1.81%	—
	DAAT	LEAST	12.99%	16.07%	1.91%	6.90%	3.44%	—
	Wand	MAXTF	12.99%	16.11%	1.94%	6.91%	3.44%	—
LM	TAAT	LEAST	28.65%	38.47%	42.76%	69.34%	59.53%	—
	MaxScore	MAXTF	33.54%	49.07%	46.36%	74.63%	61.02%	—
	DAAT	LEAST	22.32%	17.79%	14.72%	22.53%	19.20%	—
	MaxScore	MAXTF	26.29%	22.45%	18.68%	29.49%	23.18%	—
	DAAT	LEAST	23.80%	28.47%	24.88%	47.06%	40.33%	—
	Wand	MAXTF	28.45%	35.33%	31.02%	55.97%	46.32%	—
DLH13	TAAT	LEAST	25.03%	44.53%	38.63%	64.53%	58.19%	—
	MaxScore	MAXTF	32.97%	54.09%	47.54%	70.90%	59.23%	—
	DAAT	LEAST	20.15%	14.44%	9.99%	18.43%	15.08%	—
	MaxScore	MAXTF	21.36%	17.43%	13.58%	22.50%	19.14%	—
	DAAT	LEAST	20.28%	23.57%	15.99%	38.89%	32.61%	—
	Wand	MAXTF	21.76%	28.83%	22.31%	44.51%	39.96%	—

For LM and DLH13, we observe similar results, however the pruning is more difficult than for BM25. There is not a strict correlation of pruning accuracy with query length, with the only exception being GOV2 and the DAAT MaxScore strategy. However, if we compare the number of posting scored between the LEAST and MAXTF upper bounds, we make the following observations. For the GOV2 collection, the overhead of the MAXTF approximation is marked for queries with two to three terms (around 30% for LM and 22% for DLH13) but it decreases to 10% with longer queries in both cases. For the CW09B collection and the MSN query log, the query-length correlation is not present, but in any case the overhead introduced by the MAXTF approximation is usually lower than 10%. This result reflects the accuracy of the MAXTF approximation reported in Table III. In these cases too, the value of K does not have any noticable impact on our conclusions.

Comparing accuracy across the three dynamic pruning strategies, we note from Table VI that TAAT MaxScore scores more postings than the DAAT strategies. This is expected, as the DAAT strategies make more frequent pruning decisions than TAAT MaxScore, which only makes a decision after completely scoring each term. Overall,

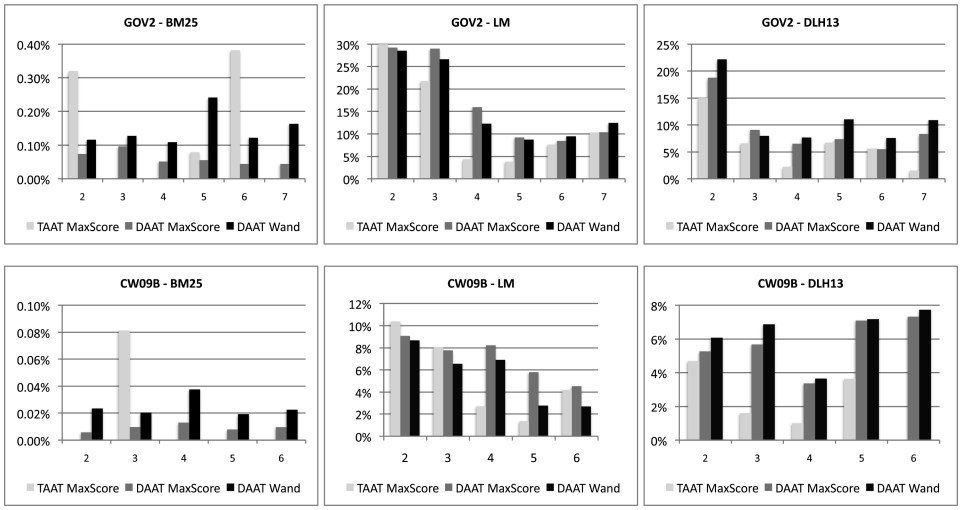

Fig. 4. Difference between the percentage of total postings scored using the MAXTF approximation and the least upper bounds, broken down by query length ($K = 1000$).

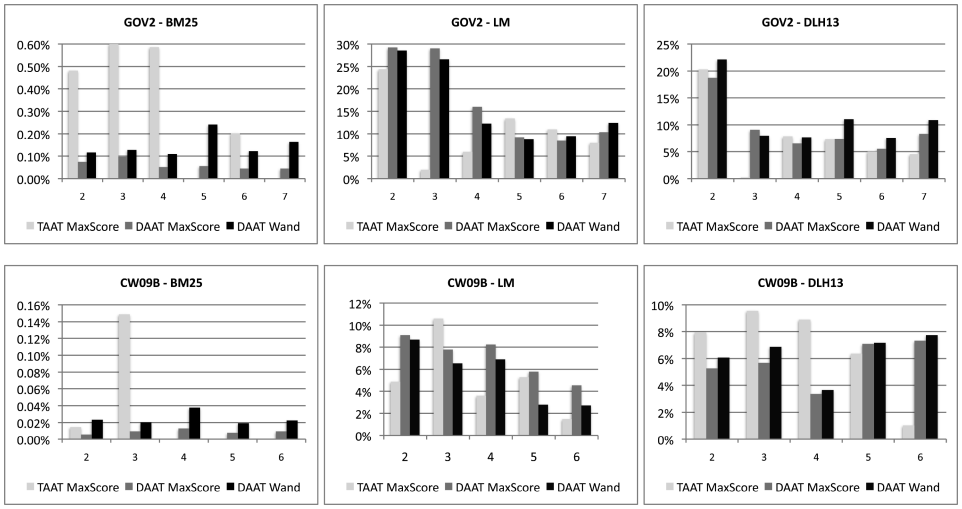

Fig. 5. Difference between the percentage of total postings scored using the MAXTF approximation and the least upper bounds, broken down by query length ($K = 20$).

DAAT MaxScore scores the least number of postings. This is due to the fact that it is focused on avoiding scoring computations, while DAAT Wand is focused on skipping read operations from disk.

While measuring the approximation impact on posting pruning, it is also possible to measure the postings associated to documents that force their way into the current top K results at that point in the processing. However, the experimental values reported by this measure are very small (i.e., 0.01–0.04% of total postings), and are exactly correlated with counting the actual number of postings scored by each dynamic pruning technique for different approximations.

Overall, our proposed MAXTF upper-bound approximation is promising, as it only scores 4%–8% more postings than the least upper bound (see query length in Table VI).

Table VI. Summary Table for Tables IV and V, Reporting Mean Percentage of Postings Scored When Using Least Upper Bound, and Increase When Using MAXTF Approximation, Across Each Experimental Dimension (Pruning Strategy, Weighting Models, K, Corpus, and Query Length)

Dimension	Variable	Mean Number of postings scored for LEAST (%)	Mean increase in postings scored for MAXTF (Δ %)
Dynamic pruning strategy	TAAT MaxScore	44.90	4.94
	DAAT MaxScore	18.88	5.18
	DAAT Wand	26.44	6.15
Weighting model	BM25	14.43	0.08
	LM	40.85	9.81
	DLH13	34.95	6.40
K	20	22.61	4.77
	1000	37.54	6.08
Corpus	GOV2	27.46	6.99
	CW09B	33.21	3.54
Query length	2 terms	31.66	8.41
	3 terms	30.28	5.67
	4 terms	25.87	4.34
	5 terms	32.40	4.57
	6 terms	31.50	4.00

However, the impact of increased postings is also likely to impact the query response time. In the next section, we validate the impact by directly timing the efficiency of the IR system using least and approximate upper bounds.

5.4. Approximation Impact on Efficiency

To further assess the efficiency of the proposed approximation and to address our third research question, we performed additional experiments to evaluate the impact of the adoption of our approximation with respect to the least upper bounds on the average query response time. We collected the query response times for the least and MAXTF approximated upper bounds using the second 500 queries from each query set—indeed, we discarded the first 500 queries to take into account any transient effects due to L2 and file caches warmup.[4] For BM25 performance loss was mostly negligible, given the small number of additional postings scored by the MAXTF approximation. Tables VII and VIII report the mean relative efficiency degradation of the MAXTF approximation with respect to the least upper bound for LM and DLH13 with $K = 1000$ and $K = 20$, respectively. Finally, to aid in the analysis of Tables VII and VIII, summary Table IX reports the mean response times, and the relative degradation for each experimental dimension (K, corpus, weighting model, dynamic pruning strategy, and query length), while varying the other dimensions.

First, we observe that GOV2 experiments show a higher efficiency overhead than CW09B experiments—for instance, in Table IX, we note that while the mean response time for GOV2 is lower than CW09B, using the MAXTF approximation is 10.3% slower than using the least upper bound. This is expected, as for the LM and DLH13 weighting models, the number of extra postings scored for GOV2 was generally than higher for CW09B in Tables IV and V above. In summary, the number of posting scores accurately reflects the response time—indeed, we found the number of postings scored and query response time over the second 500 queries to have a near-perfect correlation: Spearman's $\rho = 0.98$.

[4]However, we found a Kendall's concordance [Kendall 1955] of $W = 0.98$ between number of postings and response times across five different orderings of the query sets, suggesting that cache warmup has little impact on response times.

Table VII. Percentage Overhead of Average Query Response Time (last 500 queries) for Various Weighting Models and Dynamic Pruning Strategies ($K = 1000$)

Model	Strategy	# of Query Terms					
		2	3	4	5	6	7
GOV2							
LM	TAAT MaxScore	20.5%	8.9%	< 0.5%	5.2%	5.4%	11.9%
	DAAT MaxScore	14.2%	19.8%	8.8%	< 0.5%	0.5%	3.6%
	DAAT Wand	41.9%	46.9%	18.8%	17.5%	12%	10.7%
DLH13	TAAT MaxScore	12.1%	1.8%	2.2%	9.4%	9.0%	6.3%
	DAAT MaxScore	13.3%	7.4%	4.6%	6.0%	7.7%	6.3%
	DAAT Wand	38.4%	14.7%	13.9%	22.2%	13.8%	15.0%
CW09B							
LM	TAAT MaxScore	10.7%	< 0.5%	1.0%	3.8%	< 0.5%	—
	DAAT MaxScore	6.3%	6.0%	6.1%	3.8%	3.9%	—
	DAAT Wand	8.1%	4.1%	7.7%	< 0.5%	2.1%	—
DLH13	TAAT MaxScore	8.0%	0.5%	< 0.5%	8.1%	< 0.5%	—
	DAAT MaxScore	< 0.5%	1.1%	< 0.5%	3.1%	< 0.5%	—
	DAAT Wand	9.6%	6.6%	3.0%	7.8%	4.1%	—

Table VIII. Percentage Overhead of Average Query Response Time (last 500 queries) for Various Weighting Models and Dynamic Pruning Strategies ($K = 20$)

Model	Strategy	# of Query Terms					
		2	3	4	5	6	7
GOV2							
LM	TAAT MaxScore	40.4%	7.6%	10.3%	33.9%	31.1%	14.7%
	DAAT MaxScore	13.4%	7.9%	14.0%	15.2%	14.8%	16.1%
	DAAT Wand	40.6%	9.3%	32.0%	28.8%	29.5%	15.6%
DLH13	TAAT MaxScore	62.9%	16.1%	19.3%	22.7%	11.5%	9.5%
	DAAT MaxScore	7.9%	7.8%	10.7%	8.5%	10.8%	9.4%
	DAAT Wand	12.7%	37.1%	42.1%	29.1%	31.1%	32.2%
CW09B							
LM	TAAT MaxScore	6.3%	21.5%	11.8%	13.5%	0.5%	–
	DAAT MaxScore	< 0.5%	< 0.5%	< 0.5%	0.6%	< 0.5%	—
	DAAT Wand	16.9%	23.4%	28.8%	21.7%	17.9%	—
DLH13	TAAT MaxScore	4.1%	2.1%	22.3%	9.9%	4.3%	—
	DAAT MaxScore	< 0.5%	4.2%	6.1%	5.8%	9.8%	—
	DAAT Wand	3.6%	14.6%	29.5%	10.7%	33.7%	—

Second, from Tables VII and VIII, we observe that the performance overhead in terms of postings scored is reflected in terms of response time overhead. In particular, the overhead for two terms queries is marked in the GOV2 experiments (reaching 41.9% for DAAT and 62.9% for TAAT strategies). For CW09B, the overhead of two terms queries is reduced (16.9% for DAAT and 10.7% for TAAT strategies). From the point of view of the different dynamic pruning strategies, it is clear from Table IX that DAAT MaxScore is the least degraded strategy when using the approximation. Indeed, TAAT MaxScore suffers from the fact that even a slightly higher upper bound can force the full scoring of all postings for an additional query term, while DAAT Wand focuses on disk-access optimization and requires more complex data structures than DAAT MaxScore at runtime, and hence is not as efficient overall, particularly for longer queries.

Analyzing the results when varying the number of the top K results, it is natural that we observed a reduced overhead for $K = 1000$ with respect to $K = 20$. For the weighting models, besides having very good results for BM25 due to the closeness of our approximation to the least upper bound, for the remaining models, namely LM and DLH13, we obtain a similar performance overhead of $\sim 10\%$.

For the query lengths, we observe from Table IX that mean response time using the least upper bounds increases the number of terms in the queries. Using the

Table IX. Summary Table for Tables VII and VIII; Mean Response Time Using Least Upper Bounds and Mean Relative Degradation in Response Time for Each Dimension, When Comparing MAXTF Approximation with Least Upper Bound

Dimension	Variable	Mean response time (s) for LEAST (%)	Mean response time (s) for MAXTF	Mean relative degradation for MAXTF (Δ %)
Dynamic pruning strategy	TAAT MaxScore	3.64	3.86	6.2%
	DAAT MaxScore	1.38	1.43	4.2%
	DAAT Wand	1.54	1.73	12.6%
Weighting model	BM25	1.54	1.54	<0.5%
	LM	2.75	3.07	11.6%
	DLH13	2.71	3.00	10.7%
K	20	2.02	2.21	9.9%
	1000	2.64	2.78	5.5%
Corpus	GOV2	1.41	1.55	10.3%
	CW09B	3.25	3.39	4.5%
Query length	2 terms	3.88	4.19	8.1%
	3 terms	6.92	7.16	3.5%
	4 terms	9.16	9.39	2.5%
	5 terms	21.59	22.60	4.7%
	6 terms	24.26	24.79	2.2%

approximation results in a slight overhead in query response time (2–8%), which mostly decreases for longer queries. This is not surprising, given that more terms mean more regular pruning decisions, which provide more opportunities to prune postings. This result is mirrored by the lower increase in extra postings scored for longer queries (see Figures 4 and 5).

However, on inspection of Tables VII and VIII, we note a wide variance of increases in relative response time for different query lengths. In particular, the query terms chosen by the users will affect the response times. Some queries are easier to prune for a given dynamic pruning strategy, either because one particular chosen term is highly informative (e.g. short posting list), or because the high value documents are nearer the start of the posting lists, or because most of the postings for that term score significantly lower than the term upper bound. Hence, difficult queries are most likely to be negatively impacted by the latter reason when using approximations, as the approximate upper bound will not discriminate as well between the documents that should or should not make the top K.

Overall, we conclude that the degradation in efficiency when using the MAXTF approximation is not marked and is generally less than 10%.

5.5. Static Scores

In this section, we aim to investigate the impact of adding static scores into the retrieval process. As discussed in Section 5.1, we make use of the bounding of static scores on the retrieval process. In particular, in our experiments, the upper bound of the PageRank score for any document is $\omega = 2$.

Tables X and XI report the impact on the number of postings scored for $K = 1000$ and $K = 20$ when the PageRank static score is added to the retrieval process. Moreover, these tables are summarized in Table XII. First, as the effectiveness of the IR system has been altered by the addition of PageRank into the retrieval process, we note that the percentage of postings scored by the LEAST upper bound is different from Table IV and V.

By analyzing all three tables, we note that the results are, in general, similar to our earlier experiments that do not use PageRank. In particular, the number of postings scored for the least upper bound are typically slightly higher (on average, 14.43% for BM25 without PageRank, 19.49% with—see Tables VI and XII). If we examine each

Table X. Percentage of Postings Scored by LEAST and MAXTF Upper-Bound Approximations with PageRank Static Score, Broken Down by Query Length ($K = 1000$)

Model	Strategy	Approx.	# of Query Terms					
			2	3	4	5	6	7
GOV2								
BM25	TAAT	LEAST	51.85%	40.31%	38.25%	36.25%	40.53%	35.25%
	MaxScore	MAXTF	52.01%	40.42%	38.28%	36.29%	40.53%	37.39%
	DAAT	LEAST	27.03%	11.92%	6.71%	5.11%	4.71%	3.36%
	MaxScore	MAXTF	27.12%	12.02%	6.77%	5.17%	4.75%	3.4%
	DAAT	LEAST	36.81%	18.38%	13.27%	11.75%	11.07%	8.49%
	Wand	MAXTF	37.17%	18.67%	13.48%	11.97%	11.38%	8.62%
LM	TAAT	LEAST	95.1%	69.74%	62.33%	66.89%	67.77%	73.58%
	MaxScore	MAXTF	100%	92.98%	74.91%	72.12%	72.58%	76.29%
	DAAT	LEAST	57.91%	33.27%	27.47%	26.32%	25.96%	29.51%
	MaxScore	MAXTF	87.06%	62.29%	43.46%	35.54%	34.42%	39.88%
	DAAT	LEAST	88.47%	65.67%	52.01%	48.65%	49.68%	58.57%
	Wand	MAXTF	99.35%	89.76%	72.56%	63.25%	58.98%	66.75%
DLH13	TAAT	LEAST	88.11%	56.3%	58.33%	65.13%	66.58%	66.51%
	MaxScore	MAXTF	100%	72.68%	61.62%	67.49%	71.83%	73.15%
	DAAT	LEAST	41.06%	25.52%	21.32%	19.06%	20.42%	19.5%
	MaxScore	MAXTF	59.79%	34.59%	27.85%	26.44%	25.94%	27.84%
	DAAT	LEAST	66.5%	40.5%	37.87%	37.52%	39.74%	40.35%
	Wand	MAXTF	86.27%	55.36%	45.9%	46.07%	47.84%	52.88%
CW09B								
BM25	TAAT	LEAST	42.05%	49.88%	30.35%	40.81%	34.46%	—
	MaxScore	MAXTF	42.05%	49.88%	30.35%	40.81%	34.46%	—
	DAAT	LEAST	13.96%	17.91%	5.86%	6.34%	4.42%	—
	MaxScore	MAXTF	13.97%	17.92%	5.87%	6.35%	4.43%	—
	DAAT	LEAST	38.7%	42.2%	15.36%	34.26%	29.46%	—
	Wand	MAXTF	38.74%	42.23%	15.42%	34.3%	29.61%	—
LM	TAAT	LEAST	74.01%	82.16%	66.6%	82.02%	86.77%	—
	MaxScore	MAXTF	77.61%	84.56%	72.04%	84.01%	87.14%	—
	DAAT	LEAST	49.78%	43.15%	35.22%	49.13%	46.79%	—
	MaxScore	MAXTF	58.86%	50.94%	43.46%	54.93%	51.32%	—
	DAAT	LEAST	71.99%	73.88%	63.5%	78.03%	77.86%	—
	Wand	MAXTF	79.38%	81.36%	73.21%	82.91%	80.11%	—
DLH13	TAAT	LEAST	59.95%	71.76%	62%	82.02%	85.67%	—
	MaxScore	MAXTF	67.79%	76.31%	62.91%	82.02%	86.04%	—
	DAAT	LEAST	38.1%	29.71%	26%	37.17%	33.19%	—
	MaxScore	MAXTF	43.38%	35.4%	29.37%	44.27%	40.52%	—
	DAAT	LEAST	52.64%	57.37%	46.71%	69.22%	68.91%	—
	Wand	MAXTF	59.63%	63.25%	51.69%	72.03%	72.75%	—

of the dynamic pruning techniques in turn, we can see that DAAT MaxScore is less affected (18.88% without PageRank, 18.80% with), as it does not have to consider the upper bound of the static score during pruning. In contrast, TAAT MaxScore and DAAT Wand both score significantly more postings (44.90% to 53.12%, and 26.44% to 35.28%, respectively). This means that the use of the upper bound on the static score introduces an additional 'slackness' into pruning decisions, such that extra postings are sometimes scored unnecessarily.

However, when we consider the difference between the least upper bound and the MAXTF approximation, we note that the increases are not markedly between those without PageRank and those with (again, comparing Tables VI and XII). For instance, using the approximate upper bound on GOV2 results in 6.99% extra postings being scored without PageRank and 7.03% with. In some cases, the average number of extra postings scored using the approximation can be slightly less when using PageRank—this is probably due to the change in retrieval effectiveness rather than the tightness of the upper bounds.

Table XI. Percentage of Postings Scored by LEAST and MAXTF Upper-Bound Approximations with PageRank Static Score, Broken Down by Query Length ($K = 20$)

Model	Strategy	Approx.	# of Query Terms					
			2	3	4	5	6	7
GOV2								
BM25	TAAT	LEAST	35.87%	34.68%	27.48%	28.58%	30.59%	26.87%
	MaxScore	MAXTF	35.91%	34.68%	27.89%	28.67%	31.32%	27.22%
	DAAT	LEAST	19.19%	8.82%	4.49%	3.11%	2.76%	1.95%
	MaxScore	MAXTF	19.28%	8.83%	4.51%	3.13%	2.83%	1.96%
	DAAT	LEAST	18.83%	7.09%	4.49%	3.69%	3.86%	2.74%
	Wand	MAXTF	19.01%	7.27%	4.74%	3.81%	3.96%	2.84%
LM	TAAT	LEAST	60.45%	45.79%	52.76%	52.57%	55.44%	55.64%
	MaxScore	MAXTF	86.7%	57.92%	57.9%	64.63%	63.53%	61.69%
	DAAT	LEAST	21.66%	17.01%	12.98%	11.14%	10.85%	10.74%
	MaxScore	MAXTF	39.12%	21.88%	20.49%	19.69%	18.87%	19.07%
	DAAT	LEAST	40.24%	28.56%	28.64%	25.74%	27.47%	27.3%
	Wand	MAXTF	72.55%	41.6%	38.69%	39.92%	37.78%	46.56%
DLH13	TAAT	LEAST	57.23%	44.06%	51.75%	47.39%	52.78%	54.21%
	MaxScore	MAXTF	71.06%	45.49%	56.66%	57.26%	55.91%	57.85%
	DAAT	LEAST	21.04%	14.44%	9.64%	8.34%	9%	8.92%
	MaxScore	MAXTF	24.9%	18.73%	14.72%	11.87%	12.81%	13.66%
	DAAT	LEAST	19.78%	24.09%	18.76%	17.43%	19.9%	24.55%
	Wand	MAXTF	35%	28%	29.23%	25.28%	27.35%	30.42%
CW09B								
BM25	TAAT	LEAST	31.71%	31.62%	17.98%	21.72%	8.03%	—
	MaxScore	MAXTF	31.71%	31.62%	18.1%	21.72%	8.03%	—
	DAAT	LEAST	12.13%	8.2%	3.24%	2.25%	1.57%	—
	MaxScore	MAXTF	12.13%	9.9%	3.26%	2.25%	1.57%	—
	DAAT	LEAST	24.93%	23.38%	4.13%	15.05%	8.16%	—
	Wand	MAXTF	24.94%	23.41%	4.16%	15.07%	8.18%	—
LM	TAAT	LEAST	42.72%	52.78%	45.69%	74.02%	60.4%	—
	MaxScore	MAXTF	52.6%	62.08%	52.55%	78.44%	64.77%	—
	DAAT	LEAST	22.33%	17.79%	14.72%	22.53%	19.2%	—
	MaxScore	MAXTF	26.29%	22.45%	18.68%	29.49%	23.18%	—
	DAAT	LEAST	33.4%	37.5%	31.54%	56.05%	45.28%	—
	Wand	MAXTF	40.44%	43.41%	36.83%	63.01%	57.51%	—
DLH13	TAAT	LEAST	38.07%	59.16%	48.38%	70.97%	59.23%	—
	MaxScore	MAXTF	46.95%	62.49%	53.59%	77.73%	62.56%	—
	DAAT	LEAST	20.18%	14.44%	9.99%	18.43%	15.08%	—
	MaxScore	MAXTF	21.36%	17.43%	13.58%	22.5%	19.14%	—
	DAAT	LEAST	23.23%	29.98%	22.83%	44.23%	40.43%	—
	Wand	MAXTF	26.54%	37.05%	32.23%	51.88%	49.68%	—

Overall, we conclude that a static score can be successfully integrated into a dynamic pruning retrieval process, regardless of whether the precalculated least upper bound or an approximate upper bound on the terms scores is used.

6. PROXIMITY MODELS

Modern retrieval approaches apply not just single-term weighting models when ranking documents. In common use are proximity weighting models, which highly score the co-occurrence of pairs of query terms in close proximity to each other in documents. In this manner, the basic ranking model of an IR system (Equation (1)) is expanded as

$$\text{score}_Q(d, Q) = \omega\, S(d) + \kappa \sum_{t \in Q} \text{score}(tf_d, *_d, *_t) + \phi\; prox(d, Q),$$

for some proximity document scoring function $prox(d, Q)$, and weight ϕ.

The main approaches to integrate proximity weighting models into pruning strategies require modifications to the index structure to include information on the proximity scores upper bounds. In Schenkel et al. [2007] and Zhu et al. [2007, 2008], the authors

Table XII. Summary Table for Tables X and XI, Containing Mean Percentage of Postings Scored When Using Least Upper Bound, and Increase When Using the MAXTF Approximation, Across Each Experimental Dimension (Pruning Strategy, Weighting Models, K, Corpus, and Query Length)

Dimension	Variable	Mean number of postings scored for Factor (%)	Mean increase in postings scored for MAXTF (Δ %)
Dynamic pruning strategy	TAAT MaxScore	53.12	4.54
	DAAT MaxScore	18.80	5.21
	DAAT Wand	35.28	6.70
Weighting model	BM25	19.49	0.14
	LM	47.59	9.75
	DLH13	40.13	6.56
K	20	26.85	5.22
	1000	44.62	5.74
Corpus	GOV2	33.00	7.03
	CW09B	39.02	3.62
Query length	2 terms	42.69	7.76
	3 terms	36.92	6.27
	4 terms	29.96	4.95
	5 terms	36.63	4.53
	6 terms	35.11	3.88

detail several approaches to leveraging early termination when proximity scores are included in the ranking model. While these strategies alter the index structure (e.g. by adding term-pair inverted indices), we aim to determine how accurately the proximity scores can be upper-bounded without modifying the index structure (other than keeping-position occurrence information in the standard inverted index posting list) and exploiting the approximation obtained in Section 4. In particular, we use the sequential dependence model of Markov Random Fields (MRF) [Metzler and Croft 2005], which has been shown be to effective at modeling the proximity of query-term occurrences in documents. In MRF, the proximity score is calculated as follows:

$$prox(d, Q) = \sum_{p=(t_i,t_{i+1})\in Q} (\text{score}(pf(t_i, t_{i+1}, d, k_1), l_d, *_p) + \text{score}(pf(t_i, t_{i+1}, d, k_2), l_d, *_p)),$$

where $pf(t_i, t_{i+1}, d, k)$ represents the number of occurrences of the tuple of sequential query terms (t_i, t_{i+1}) occurring in document d in windows of size k (abbreviated as pf_d). Following Metzler and Croft [2005], we set $\phi = 0.1$, and $k_1 = 2$, and $k_2 = 8$ to account for the proximity of two terms as an exact phrase, and proximity at distance 8, respectively. score$(pf_d, l_d, *_p)$ is implemented using Dirichlet language modeling, as per Equation (8), where pair frequency pf_d takes the role of term frequency tf_d. However, in contrast to term weighting, in proximity weighting, it is common to assume a constant frequency for the pair in the collection [Macdonald and Ounis 2010].[5] It then follows that MRF can easily be calculated in a DAAT retrieval strategy without the need for multiple passes over the postings lists.

As score$(pf_d, l_d, *_p)$ is a Dirichlet language model, the result of Theorem 2 holds—in particular score$(pf_d, l_d, *_p)$ is strictly monotonic in pf_d. The question, then, is how to estimate an upper bound on pf_d without special indexing support. Given that a tuple (t_i, t_{i+1}) cannot occur in any document more frequently than the smallest term frequency of the two terms in the document, an upper bound on pf_d for all occurrences

[5]As implemented by the authors of MRF in the Ivory retrieval system; see http://www.umiacs.umd.edu/~jimmylin/ivory.

Table XIII. Average Query Response Time (Last 500 Queries, in Seconds) for the Application of MRF ($K = 20$)

Strategy	# of Query Terms						Mean
	2	3	4	5	6	7	
GOV2							
DAAT FULL	1.27	3.24	5.80	7.71	11.52	15.02	4.79
DAAT FULL + MRF	1.28	3.25	5.97	8.25	14.13	17.10	5.10
DAAT MAXSCORE	0.69	1.70	3.18	4.28	7.09	8.61	2.63
DAAT MAXSCORE + MRF	0.81	1.97	3.71	5.06	8.54	10.74	3.11
CW09B							
DAAT FULL	4.59	7.18	9.49	20.43	24.96	—	6.35
DAAT FULL + MRF	4.85	7.76	10.58	24.14	30.43	—	7.07
DAAT MAXSCORE	3.21	5.02	6.43	15.16	16.94	—	4.57
DAAT MAXSCORE + MRF	4.01	6.20	7.73	19.95	22.25	—	5.76

of two terms can be calculated as

$$pf_{max} = \min\left(\max_{d \in L(t_i)} tf_d, \max_{d \in L(t_j)} tf_d\right).$$

This permits efficient proximity calculation on an index using positions without the need for any additional index structures, unlike Schenkel et al. [2007] and Zhu et al. [2007, 2008].

We adapted the DAAT MAXSCORE strategy to include proximity scores in the scoring function.[6] In this way, both proximity and term scores contribute at the same time to determine the current threshold of the top-K documents and the upper bound for the currently scored document. In the following, we experiment to determine if the MAXTF approximate can be used to prune pair postings for the MRF approach, when combined with LM and the DAAT MAXSCORE dynamic pruning strategy. The experimental setup is unchanged from that described in Section 5; however, the postings in the inverted index now contain occurrence position information, in the form of Elias-unary encoded position gaps. Moreover, in contrast to Section 5, we assert that it is infeasible to precalculate the least upper bounds for each pair of terms in the index. Hence, this experiment cannot obtain relative degradation in efficiency with respect to the least upper bounds, nor the number of extra pair postings being needlessly scored.

Table XIII shows the average query response time for DAAT FULL and DAAT MAXSCORE strategies, with and without the application of MRF, for $K = 20$. As in Section 5.4, the response times are reported for the second 500 queries, as the first 500 queries are omitted as cache warmup. From the results, we first note that the response times for the DAAT MAXSCORE strategy without MRF are higher than those reported earlier in Table IX. This is expected because the inverted index size is markedly increased by the presence of occurrence-position information (8 GB to 34 GB for GOV2, 20 GB to 78 GB for CW09B). Nevertheless, DAAT MAXSCORE is faster than the exhaustive DAAT FULL strategy.

Next, adding MRF to a full DAAT strategy can noticeable decrease efficiency (e.g., 4.79 to 5.10 s across all 500 queries with the GOV2 collection). Moreover, MRF has more impact on efficiency as query length increases. This is expected, as the number of pairs of query terms also increases. However, as expected, DAAT MAXSCORE is faster than the exhaustive DAAT FULL strategy. Moreover, MRF can successfully be integrated into DAAT MAXSCORE, such that not every posting is scored, and, hence, is more efficient that DAAT FULL + MRF. This suggests that the MAXTF approximation is suitable for use in proximity as well, without the need for special index structures such as term pair posting lists [Schenkel et al. 2007; Zhu et al. 2007, 2008].

[6]See Tonellotto et al. [2010] for a study of dynamic pruning strategies for proximity models.

7. CONCLUSIONS

In this work, we analyzed the problem of obtaining upper bounds for weighting model scores by approximation, instead of precalculation. We then showed how upper bounds based on maximal term frequency could be proven safe for various modern weighting models, by formulating a constrained maximization problem. By relaxing the integral constraint of the problem, we were able to derive a novel yet provably effective approximation for several weighting models from different families. The accuracy and efficiency of all upper bounds were then empirically tested using several dynamic pruning strategies using many queries on two large-scale TREC test collections. Moreover, we investigated the effect of introducing the PageRank static score into the dynamic pruning retrieval approach, and the applicability of the proposed approximation to the Markov Random Field proximity model.

The experiments within this paper show that efficient retrieval systems can be obtained for various weighting models without the need to precalculate term upper bounds for each weighting model. For instance, using approximate upper bounds results in only 2%–8% degradations in mean query response time compared to using the least upper bound.

In Section 2, we motivated the choice of disjunctive query processing in our experiments through effectiveness evidence in the literature. Nevertheless, conjunctive processing allows additional efficiency improvements. In the future, we will compare and contrast the suitability of our upper-bound approximations for both conjunctive and disjunctive query processing. Moreover, in Section 5.2, we noted that our upper-bound approximations only overestimated the least upper bound by 3–20%. In future work, we will reexamine the impact on both efficiency and effectiveness as the upper bound is reduced. Furthermore, in Section 5.4, we noted that some queries are more difficult than others of the same length for pruning—we will investigate if it is possible to characterize such queries, or even predict them before retrieval starts. Finally, we also intend to examine the use of upper-bound approximations when dynamic pruning strategies are applied in document-partitioned distributed retrieval systems.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and the editor for their useful comments, which have helped improving the quality of this article.

REFERENCES

ALTINGOVDE, I. S., OZCAN, R., AND ULUSOY, O. 2009. Exploiting query views for static index pruning in web search engines. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*. J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi, and A. Chowdhury, Eds., ACM Press, New York, NY, 1951–1954.

AMATI, G. 2006. Frequentist and bayesian approach to information retrieval. In *Proceedings of the 28th European Conference on IR Research*, M. Lalmas, A. MacFarlane, S. M. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, Eds. Springer, Berlin, Germany, 13–24.

ANAGNOSTOPOULOS, A., BRODER, A. Z., AND CARMEL, D. 2005. Sampling search-engine results. In *Proceedings of the 14th International Conference on World Wide Web*. A. Ellis and T. Hagino, Eds., ACM Press, New York, NY, 245–256.

ANH, V. N. AND MOFFAT, A. 2001. Pruned query evaluation using precomputed impact scores. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*. A. Coden, E. W. Brown, and S. Srinivasan, Eds., ACM Press, New York, NY.

BENDERSKY, M., CROFT, W. B., AND SMITH, D. A. 2009. Two-stage query segmentation for information retrieval. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. J. Allan, J. A. Aslam, M. Sanderson, C. Zhai, and J. Zobel, Eds. ACM Press, New York, NY, 810–811.

BLANCO, R. 2008. Index compression for information retrieval systems. Ph.D. dissertation. University of A Coruna, A Coruna, Spain.

BRODER, A. Z., CARMEL, D., HERSCOVICI, M., SOFFER, A., AND ZIEN, J. 2003. Efficient query evaluation using a two-level retrieval process. In *Proceedings of the 12th International Conference on Information and Knowledge Management*. H. Paques, L. Liu, and D. Grossman, Eds. ACM Press, New York, NY, 426–434.

BUTTCHER, S., CLARKE, C. L. A., AND SOBOROFF, I. 2007. The TREC 2006 Terabyte track. In *Proceedings of the 15th Text Retrieval Conference*. E. M. Voorhees and L. P. Buckland, Eds. NIST, Gaithersburg, MD.

CLARKE, C. L., CRASWELL, N., AND SOBOROFF, I. 2010. Overview of the TREC 2009 Web track. In *Proceedings of the 18th Text Retrieval Conference*. E. M. Voorhees and L. P. Buckland, Eds. NIST.

CRASWELL, N., FETTERLY, D., AND NAJORK, M. 2011. The power of peers. In *Proceedings of the 33rd European Conference on IR Research: Advancesin Information Retrieval*. P. Clough, C. Foley, C. Gurrin, G. J. Jones, W. Kraaij, H. Lee, and V. Murdoch, Eds. Springer, Berlin, Germany.

CRASWELL, N., JONES, R., DUPRET, G., AND VIEGAS, E. Eds. 2009. In *Proceedings of the Web Search Click Data Workshop at WSDM'09*. ACM Press, New York, NY.

CROFT, W. B., METZLER, D., AND STROHMAN, T. 2009. *Search Engines—Information Retrieval in Practice*. Addison-Wesley, Reading, MA.

FAGIN, R., LOTEM, A., AND NAOR, M. 2003. Optimal aggregation algorithms for middleware. *J. Comput. Syst. Sci. 66,* 4, 614–656.

FANG, H., TAO, T., AND ZHAI, C. 2004. A formal study of information retrieval heuristics. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval*. M. Sanderson, K. Järvelin, J. Allan, and P. Bruza, Eds. ACM Press, New York, NY, 49–56.

GENG, X., LIU, T.-Y., QIN, T., ARNOLD, A., LI, H., AND SHUM, H.-Y. 2008. Query-dependent ranking using k-nearest neighbor. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*. S.-H. Myaeng, D. W. Oard, F. Sebastiani, T.-S. Chua, and M.-K. Leong, Eds. ACM Press, New York, NY, 115–122.

HE, B. AND OUNIS, I. 2004. A query-based pre-retrieval model selection approach to information retrieval. In *Proceedings of the 7th International Conference on Computer-Assisted Information Retrieval*. C. Fluhr, G. Grefenstette, and W. B. Croft, Eds. CID, 706–719.

JONGEN, H. T., MEER, K., AND TRIESCH, E. 2004. *Optim. Theo.* Springer, Berlin, Germany.

KANG, I.-H. AND KIM, G. 2003. Query type classification for Web document retrieval. In *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval*. C. Clarke, G. Cormack, J. Callan, A. Smeaton, and D. Hawking, Eds. ACM Press, New York, NY, 64–71.

KENDALL, M. G. 1955. *Rank Correlation Methods* 2nd Ed. Charles Griffin & Company Limited, London, UK.

LACOUR, P., MACDONALD, C., AND OUNIS, I. 2008. Efficiency comparison of document matching techniques. In *Proceedings of the Efficiency Issues in Information Retrieval Workshop at ECIR*. R. Blanco and F. Silvestri, Eds.

LI, X., LI, F., JI, S., ZHENG, Z., CHANG, Y., AND DONG, A. 2009. Incorporating robustness into Web ranking evaluation. In *Proceedings of the 18th International ACM Conference on Information and Knowledge Management*. J. Huang, N. Koudas, G. Jones, X. Wu, K. Collins-Thompson, and A. An, Eds. ACM Press. New York, NY, 2007–2010.

LIU, T.-Y. 2009. Learning to rank for information retrieval. *Found. Trends Inform. Retriev. 3,* 3, 225–331.

MACDONALD, C. AND OUNIS, I. 2010. Global statistics in proximity weighting models. In *Proceedings of Web N-gram 2010 Workshop at SIGIR*. C. Zhai, D. Yarowsky, E. Viegas, K. Wang, and S. Vogel, Eds.

METZLER, D. AND CROFT, W. B. 2005. A Markov random field model for term dependencies. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval.* R. A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, and J. Tait, Eds. ACM Press, New York, NY, 472–479.

MOFFAT, A. AND ZOBEL, J. 1996. Self-indexing inverted files for fast text retrieval. *Trans. Inform. Syst. 14,* 4, 349–379.

MOURA, E. S. D., SANTOS, C. F. D., ARAUJO, B. D. S. D., SILVA, A. S. D., CALADO, P., AND NASCIMENTO, M. A. 2008. Locality-based pruning methods for web search. *Trans. Inform. Syst. 26*, 9, 1–9:28.

PERSIN, M. 1994. Document filtering for fast ranking. In *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval*. W. B. Croft and C. J. van Rijsbergen, Eds. ACM Press, New York, NY.

ROBERTSON, S. E., WALKER, S., HANCOCK-BEAULIEU, M., GULL, A., AND LAU, M. 1992. Okapi at TREC. In *Proceedings of the 1st Text REtrieval Conference.* D. K. Harman, Ed. NIST Special Publication, vol. 500-207. NIST.

SCHENKEL, R., BROSCHART, A., HWANG, S., THEOBALD, M., AND GATFORD, M. 2007. Efficient text proximity search. In *Proceedings of String Processing and Information Retrieval*. N. Ziviani and R. A. Baeza-Yates, Eds. Springer, Berlin, Germany, 287–299.

SILVERSTEIN, C., HENZINGER, M., MARAIS, H., AND MORICZ, M. 1998. Analysis of a very large AltaVista query log. Tech. rep. 1998-014. Digital SRC, Palo Alto, CA.

SKOBELTSYN, G., JUNQUEIRA, F., PLACHOURAS, V., AND BAEZA-YATES, R. 2008. Resin: a combination of results caching and index pruning for high-performance web search engines. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, New York, NY, 131–138.

TAYLOR, M., ZARAGOZA, H., CRASWELL, N., ROBERTSON, S., AND BURGES, C. 2006. Optimization methods for ranking functions with multiple parameters. In *Proceedings of the 15th International ACM Conference on Information and Knowledge Management*. P. S. Yu, V. J. Tsotras, E. A. Fox, and B. Liu, Eds. ACM Press, New York, NY, 585–593.

TONELLOTTO, N., MACDONALD, C., AND OUNIS, I. 2010. Efficient dynamic pruning with proximity support. In *Proceedings of the 8th Workshop on Large-Scale Distributed Systems for Information Retrieval at SIGIR*. R. Blanco, B. B. Cambazoglu, and C. Lucchese, Eds. CEUR Workshop Proceedings, vol. 630.

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

XIANG, B., JIANG, D., PEI, J., SUN, X., CHEN, E., AND LI, H. 2010. Context-aware ranking in web search. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. F. Crestani, S. Marchand-Maillet, H.-H. Chen, E. N. Efthimiadis, and J. Savoy, Eds. ACM Press, New York, NY, 451–458.

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.

ZHU, M., SHI, S., LI, M., AND WEN, J.-R. 2007. Effective top-k computation in retrieving structured documents with term-proximity support. In *Proceedings of the 16th International ACM Conference on Information and Knowledge Management*. M. J. Silva, A. H. F. Laender, R. A. Baeza-Yates, D. L. McGuinness, B. Olstad, Ø. H. Olsen, and A. O. Falcão, Eds. ACM Press, New York, NY, 771–780.

ZHU, M., SHI, S., YU, N., AND WEN, J.-R. 2008. Can phrase indexing help to process non-phrase queries? In *Proceedings of the 17th International ACM Conference on Information and Knowledge Management*. J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi, and A. Chowdhury, Eds. ACM Press, New York, NY, 679–688.

ZHU, Z. A., CHEN, W., WAN, T., ZHU, C., WANG, G., AND CHEN, Z. 2009. To divide and conquer search ranking by learning query difficulty. In *Proceedings of the 18th International ACM Conference on Information and Knowledge Management*. J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi, and A. Chowdhury, Eds. ACM Press, New York, NY, 1883–1886.

Received April 2010; revised November 2010, May 2011; accepted June 2011