Upper Bound Approximations for BlockMaxWand

Craig Macdonald
University of Glasgow
Glasgow, Scotland, UK
craig.macdonald@glasgow.ac.uk

Nicola Tonellotto
ISTI-CNR
Pisa, Italy
nicola.tonellotto@isti.cnr.it

ABSTRACT

BlockMaxWand is a recent advance on the Wand dynamic pruning technique, which allows efficient retrieval without any effectiveness degradation to rank K. However, while BMW uses docid-sorted indices, it relies on recording the upper bound of the term weighting model scores for each *block* of postings in the inverted index. Such a requirement can be disadvantageous in situations such as when an index must be updated. In this work, we examine the appropriateness of upper-bound approximation – which have previously been shown suitable for Wand– in providing efficient retrieval for BMW. Experiments on the ClueWeb12 category B13 corpus using 5000 queries from a real search engine's query log demonstrate that BMW still provides benefits w.r.t. Wand when approximate upper bounds are used, and that, if approximations on upper bounds are tight, BMW with approximate upper bounds can provide efficiency gains w.r.t. Wand with exact upper bounds, in particular for queries of short to medium length.

KEYWORDS

Wand, BlockMaxWand, upper-bounds approximations

1 INTRODUCTION

The efficiency of a search engine is important, for example to ensure user satisfaction (users will not wait a long time for results), and also to minimise the resources that must be deployed by the search engine (number of servers needed to ensure low response times). A key factor in ensuring such low response time is the layout and traversal strategies of the inverted index underlying the search engine. In this paper, we are concerned with the efficient traversal of docid-sorted inverted index posting lists, as these are more commonly deployed in industry [4], rather than impact sorted postings lists.

Among techniques, the Wand technique [1], and the more recent variant BlockMaxWand (BMW) [6] are advantageous to deploy, as they enable effecient retrieval of K documents without degrading effectiveness to rank K (also known as safe-to-rank K). In particular, Wand and BMW determine the query terms that must be matched for the next document to be retrieved, based on upper bounds of the scores of the query terms, and the score of the current K-th ranked document. Efficiency is therefore enhanced as the decompression of postings and the scoring of documents that cannot make the current K ranked documents are *skipped*. The advance offered by BMW is that upper bounds are calculated for blocks of postings, offering tighter upper bounds than a single upper bound for the entire posting list, and hence more skipping is achieved.

Upper bounds for a given weighting model are typically calculated by pre-scoring all postings for each query term in the inverted index. However, such pre-calculated upper bounds have disadvantages [8], for instance that they are sensitive to changes in weighting model scores, as might be caused by additions/deletions to the index, or by changes to the weighting model parameters. In [8], the authors proposed *approximations* for upper bounds for Wand, applicable to various weighting models. Such approximations are "less tight" than the exact (empirically-derived) calculated upper bounds, but only require more basic statistics such as the maximum within-document term frequency in each posting list.

However, no previous work has addressed the application of upper bounds for BMW. Hence, in this work, our central contribution is to experiment to address a central research question: are approximations of UBs good enough for efficient retrieval using BMW? The remainder of this paper is as follows: Section 2 provides an overview of the Wand and BMW techniques; Section 3 describes the calculation of exact and approximate upper bounds on term weighting score contributions. In Section 4 we demonstrate and analyse the applicability of approximate upper bounds for BMW. Section 5 provides concluding remarks.

2 QUERY PROCESSING

In document-at-a-time (DAAT) query processing, the query term postings lists are processed in parallel keeping them aligned by docid. The score of each document is computed fully by considering the contributions of all query terms $t \in Q$ before moving to the next document. However, processing queries exhaustively with DAAT can be very inefficient, and therefore various techniques to enhance retrieval efficiency have been proposed, by dynamically pruning docids that are unlikely to be retrieved. Among them, the most popular today is Wand [1]. This processing strategy uses additional information for each term in the form of its maximum score contribution, or *upper bound* $\sigma(t)$, thus allowing to skip large segments of posting lists if they only contain terms whose sum of maximum scores is smaller than the scores of the top K documents found up to that point. Wand relies on upper-bounding the contribution that each term can give to the overall document score, allowing to skip whole ranges of docids [8].

Wand employs a *global* per-term upper bound, that is, the maximum score among *all* documents in a given term's posting list. Such maximum score could be significantly larger than the typical score contribution of that term, in fact limiting the opportunities to skip large amounts of documents. To tackle this problem, Ding and

ICTIR '17, October 1–4, 2017, Amsterdam, Netherlands.

ISBN 978-1-4503-4490-6/17/10…$15.00
DOI: https://doi.org/10.1145/3121050.3121094

Suel [6] proposed to augment the inverted index data structures with additional information to store more accurate upper bounds: at indexing time each posting list is split into consecutive blocks of constant size, e.g. 128 postings per block. For each block B the score upper bound $\sigma^B(t)$ is stored, together with largest docid of each block. These *block* term upper bounds can then be exploited by adapting existing algorithms such as Wand to make use of the additional information. The resulting algorithm is BlockMaxWand (BMW) [6]. The authors reported an average query response time reduction of BMW compared to Wand of 64% – 67%. Experiments in [5] reported a reduction of 66% by BMW with respect to Wand. A more recent work [2] explored the performance of BMW compared to Wand on different document collection and different query logs. They reported average reductions up to only 26%. However, for long queries and large collections, Wand outperforms BMW, because of its complex logic for skipping blocks using block upper bounds.

In the following, we discuss how recent advances in determine *approximate* upper bounds can be applied for both Wand and BMW.

3 DEFINING UPPER BOUNDS

As shown in the previous section, both Wand and BMW rely on upper bounds for the maximum contribution of the weighting model for each query term, i.e. $\sigma(t)$ for an entire posting list, or $\sigma^B(t)$ for a block of postings B. In the following, we discuss both the classical empirical evaluation of *exact* upper bounds – by pre-scoring of the index – as well as recent advances in *approximate* upper bounds.

3.1 Exact Upper Bounds

Classically, such upper bounds can be calculated exactly by pre-scoring of each term's postings list $p(t)$:

$$\sigma_{\text{EXACT}}(t) = \max_{d \in p(t)} w(tf_d, l_d) \tag{1}$$

for some weighting model $w(\cdot, \cdot)$ calculated using the within-document term frequency tf_d and length of document d. These upper bounds are then stored within an augmented inverted index data structure.

However, as highlighted in Section 1, the exact pre-calculation of $\sigma_{\text{EXACT}}(t)$ has some disadvantages:

(1) Adaptation of the weighting model, or its hyper-parameters;
(2) Adaptation of the index, e.g. adding or removing documents, thereby changing global statistics of the index (number of documents, average document length);
(3) Adaptation of a given term's posting list, e.g. adding or removing documents, thereby changing the statistics of the term (e.g. IDF).

Given these disadvantages, the use of pre-calculated exact upper bounds that are stored within an augmented inverted index may not be suitable for some retrieval environments. For this reason, Macdonald et al. [8] investigated the use of approximate upper bounds for Wand. Below, we discuss approximate upper bounds, and their application to Wand and BMW.

3.2 Approximate Upper Bounds

Approximate upper bounds [8] are upper bounds $\sigma_{\text{APPROX}}(t)$ that can be calculated based on raw index statistics. They are designed to be *safe*, i.e. $\sigma_{\text{APPROX}}(t) \geq \sigma_{\text{EXACT}}(t)$, which means that given in retrieving K documents, effectiveness to rank K will not be negatively impacted (also known as safe-to-rank K). Moreover, the *accuracy* of the approximate upper bounds – the extent that they over-estimate the actual exact upper bound is important: widely inaccurate upper bounds will lead to the unnecessary scoring of documents that could never make the top K retrieved set as their approximate scoring was over-estimated. Hence, the absolute error $\sigma_{\text{APPROX}}(t) - \sigma_{\text{EXACT}}(t)$ should be minimised.

To derive approximate upper bounds for weighting models such as BM25, Dirichlet Language Modelling (LM), and DLH13 from the Divergence from Randomness framework, Macdonald et al. [8] proposed a methodology based on partial differentiation of the weighting models w.r.t. term frequency tf_d and document length l_d.

Indeed, as weighting models are typically monotonically increasing in tf_d (this was characterised as TFC1 in the formalised heuristics identified by Fang et al. [7]), an upper bound is typically found at (or just before) tf_{max}, where $tf_{\text{max}} = \max_{d \in p(t)} tf_d$. Moreover, as longer documents have lower scores (due to document length normalisation, denoted as LNC1 in [7]), for all documents in a posting list, l_d cannot be less than tf_d. Thus an approximate upper bound that is appropriate for a number of weighting models is:

$$\sigma_{\text{APPROX}}(t) = w(tf_{\text{max}}, tf_{\text{max}} + \epsilon) \tag{2}$$

where ϵ is a small number, required for some weighting models that are not defined when $l_d = tf_d$; $\epsilon = 0$ for BM25 and Dirichlet LM, and $\epsilon = 1$ for DLH13.

As is clear from Equation (2), approximate upper bounds can be easily obtained for models such as BM25 based on storing tf_{max} alone, a statistic for each term that can be easily calculated and stored within the lexicon structure of the inverted index. It does not require knowledge of the collection's statistics, nor the weighting model hyper-parameter settings that will be applied at retrieval time, and can be easily updated when new documents are added to a term's posting list.

Within the empirical studies reported in [8], approximate upper bounds were found to be suitable for Wand and the simpler MaxScore dynamic pruning technique, but no work has investigated their applicability to the more complex BMW technique, which relies on upper bounds calculated for each block B of postings. Indeed, the central aim of this work is to investigate the usability of approximate upper bounds for blocks in the context of BMW, i.e. $\sigma^B_{\text{APPROX}}(t)$ calculated as per Equation (2), but using the maximum frequency observed in the block of postings, tf^B_{max}. Like those reported in [8] for Wand, our experiments show that the approximations can be used for BMW, but cannot match the efficiency of exact upper bounds. Approximate upper bounds, being greater than exact upper bounds, limit the skipping abilities of BMW, forcing more blocks to processed because their approximate contributions would beat the current top K documents threshold. Nevertheless, we will show that they allow to improve over the efficiency of Wand when using exact upper bounds.

4 EXPERIMENTS

Motivated by the unknown applicability of approximate upper bounds for BMW, in the following, we experiment to address two research questions:

Table 1: Mean query times (in ms) for different weighting models with both exact and approximate upper bounds (denoted ✗ and ✓, resp.). Percentage reductions are shown for BMW w.r.t. Wand, (denoted Δs), and of BMW with approximate upper bounds w.r.t. Wand with exact upper bounds (Γs).

Model	Approx.	$K = 20$				$K = 1000$			
		Wand	BMW	Δ(%)	Γ(%)	Wand	BMW	Δ(%)	Γ(%)
BM25	✗	58.67	38.71	34.02	16.02	113.06	88.46	21.76	8.55
	✓	59.18	49.27	16.75		113.91	103.39	9.24	
LM	✗	172.52	68.72	60.17	44.66	285.00	130.10	54.35	44.10
	✓	217.41	95.47	56.09		359.42	159.31	55.68	
DLH13	✗	139.67	74.42	46.72	-17.40	235.42	137.68	41.52	-4.34
	✓	167.62	163.97	2.18		272.23	245.63	9.77	

RQ1: What is the impact of upper bound approximations on BMW in terms of efficiency?

RQ2: Can we obtain efficiency benefits when using upper bound approximations with BMW w.r.t. Wand when using exact upper bounds?

In the remainder of this section, we define the experimental setup under which our experiments are conducted and we report the results and analysis addressing our two research questions.

All of our experiments are conducted on the TREC ClueWeb12 category B13 corpus[1], which consists of 50M Web documents. We index all 50M documents of the ClueWeb12 corpus using the Terrier IR platform [9], removing stopwords and applying Porter stemming. Our index is compressed using Elias-Fano encoding provided in [11], widely considered to be the state-of-the-art in terms of fast decompression. For the block upper bounds, we assume the standard block size of 128 postings.

For retrieval, we follow best practices in sampling a significant number of queries from a real search engine, namely 5,000 random queries from the MSN 2006 query log [3]. We conduct efficiency timings using a machine equipped with 32 GB RAM and an 8-core Intel i7-4770K processor. The entire index is loaded in memory. All experiments are performed on a single core. While the resulting response times using a single machine for retrieval are marginally higher than would be expected for interactive retrieval in a deployed Web search engine, following previous work [12], this does not detract from the generality of the findings, and avoids the complexities of performing experiments in a distributed retrieval environment.

Table 1 reports the mean response times, in milliseconds for Wand and BMW for $K = 20$ and $K = 1000$, for BM25, Dirichlet LM and DLH13 weighting models, when exact or approximate upper bounds are used. We also note the percentage reduction in mean response times of BMW vs. Wand, denoted Δ(%), and of BMW with approximate upper bounds w.r.t. Wand with exact upper bounds, denoted with Γ(%).

We firstly compare BMW with Wand when using exact upper bounds. Indeed, on analysing Table 1, observe that BMW with exact upper bounds provides clear improvements in mean query times for all weighting models, with greater benefits when K is smaller. In particular, LM obtains reductions in mean response time, which are > 50%. This is confirmed by the reduction on the total number of postings processed by BMW. These results confirm the findings in [10], where the authors analysed the performance of BMW and Wand in terms of number of processed documents. Indeed, in line with our results, they reported that BMW only marginally improved the performance over Wand for BM25, while BMW markedly boosted the performance when using LM for scoring.

Next, we consider the approximate upper bounds, and observe that using the approximate upper bounds increases the response times of both Wand (as expected from [8]) and also BMW. Moreover, the benefits of BMW over Wand are reduced when approximate upper bounds are used in place of exact upper bounds, both in terms of mean query response times and number of processed postings (e.g. for BM25, $K = 20$, BMW reduces response times by 34% compared to Wand for exact upper bounds, and only 17% for approximate upper bounds). Nevertheless, clear benefits w.r.t. the corresponding Wand processing with approximate upper bounds are still present when documents are evaluated with LM (e.g. for $K = 20$, BMW reduces response times by 60% compared to Wand for exact upper bounds, and 56% for approximate upper bounds). However, when documents are evaluated with DLH13, BMW with approximate upper bounds is marginally worse than Wand with exact upper bounds, with higher losses when K value is small. This loss can be explained by Figure 1, which reports the distribution of the absolute difference between approximate and exact upper bounds for all the blocks associated with query terms, for the different weighting models. BM25 exhibits the best error distribution due to the saturating effect of the Robertson's TF component in BM25 (TFC2 in [7]), the IDF component is dominant for large values of $tf^B_{\max}$. Hence its benefits are limited since the block upper bounds are similar in magnitude to the corresponding term upper bound. For LM, the block upper bounds are not concentrated towards the corresponding term upper bound [10], and the error distribution is skewed towards a small percentage of normalised absolute error. The absolute errors reported for DLH13 are relatively larger than the corresponding errors for other weighting models, i.e. the approximate upper bounds for DLH13 are significantly larger than the corresponding exact upper bounds, causing a large number of blocks to be accessed during query processing that do not contain documents that are retrieved in the final top K set.

Hence, regarding RQ1, we conclude that the use of approximate upper bounds with BMW provides a relatively small performance loss with BM25 and LM, while for DLH13 the upper bound approximations cause a reasonable loss in efficiency w.r.t. exact upper bounds. Nevertheless, BMW still provides benefits w.r.t. Wand when approximate upper bounds are used.

Next, we address RQ2, by comparing the efficiency of BMW with approximate upper bounds versus the efficiency of Wand with exact upper bounds. In doing so, we also make use of Figure 2, which reports the mean, median and errors bars of query times for Wand (with exact upper bounds) and BMW (with exact and approximate upper bounds) for multi-term queries ($K = 20$), broken down by number of terms, for different weighting models. For BM25, BMW with approximate upper bounds provides clear benefits for 2 and 3 terms queries w.r.t. Wand with exact upper bounds, and for LM clear benefits are present also for queries with more terms. As

[1] http://lemurproject.org/clueweb12/

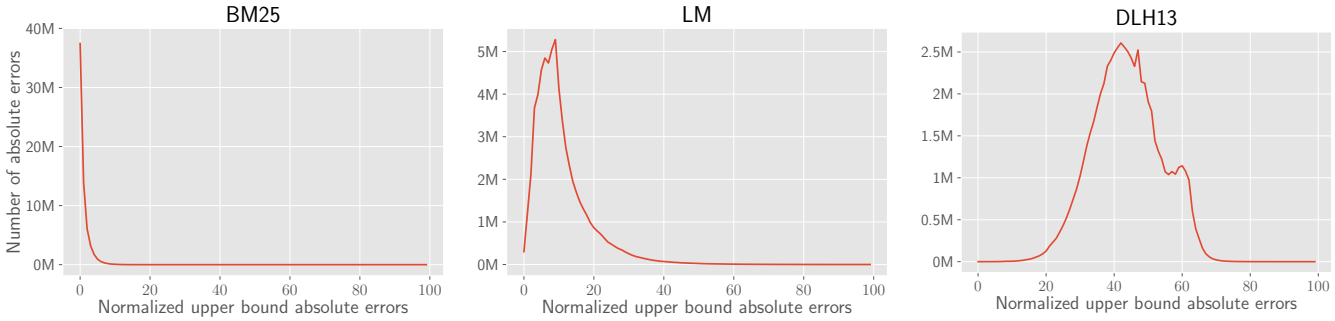

Figure 1: Distribution of block upper bound absolute errors for all the blocks associated with query terms, for different weighting models.

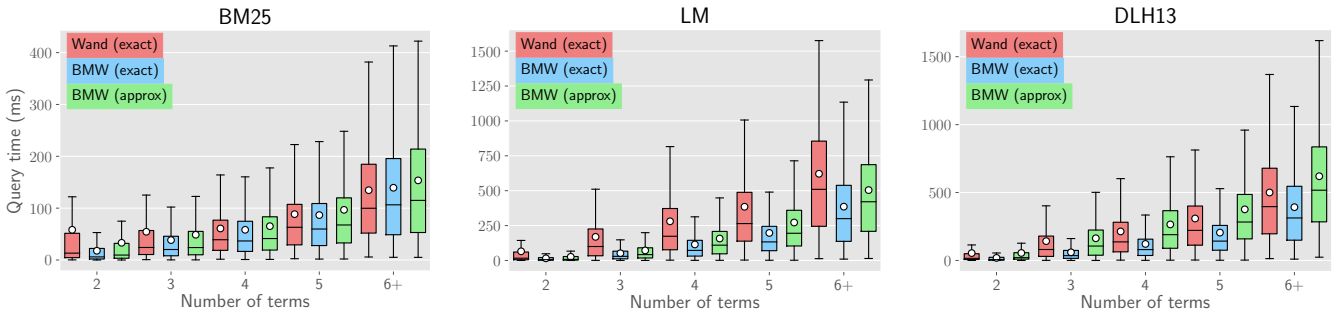

Figure 2: Mean, median and error bars (in ms) for Wand (with exact upper bounds) and BMW (with exact and approximate upper bounds) for multi-term queries ($K = 20$), broken down by number of terms, for different weighting models.

reported in Table 1 (Γ column), the overall percentage reduction of BMW with approximate upper bounds w.r.t. Wand with exact upper bounds is 8% – 16% for BM25, with larger improvements for the smaller K value, and above 40% for LM, regardless of the value of K. However, DLH13 suffers from the aforementioned approximation looseness, hence it cannot compete with BMW using exact upper bounds. Overall, for RQ2, we conclude that, if approximations of the upper bounds are sufficiently tight, BMW with approximate upper bounds can provide efficiency benefits w.r.t. Wand with exact upper bounds, in particular for queries of short or medium lengths. This is also apparent from Table 1, where we observe BMW is more sensitive to the accuracy of the upper bounds than Wand–indeed, for DLH13, $K = 20$, using approximate upper bounds only slightly degrade the efficiency of Wand (139 → 167 ms), it more than doubles the response time of BMW (74 → 163 ms). This highlights the importance of tight upper bounds approximations on the resulting efficiency of BMW.

5 CONCLUSIONS

In this paper, we demonstrated the applicability of approximate upper bounds to the BMW, which can result in marked benefits to efficiency compared to Wand using exact upper bounds (up to 44% in the case of LM). This ensures that efficient but safe retrieval can be attained in scenarios where exact upper bounds cannot be maintained. However, our results also provide insight into the importance of the tightness of the approximate upper bounds for efficient BMW, and how this varies across different weighting models.

REFERENCES

[1] Andrei Z. Broder, David Carmel, Michael Herscovici, Aya Soffer, and Jason Y. Zien. 2003. Efficient query evaluation using a two-level retrieval process. In *CIKM*. 426–434.

[2] Matt Crane, J. Shane Culpepper, Jimmy Lin, Joel Mackenzie, and Andrew Trotman. 2017. A Comparison of Document-at-a-Time and Score-at-a-Time Query Evaluation. In *WSDM*. 201–210.

[3] Nick Craswell, Rosie Jones, Georges Dupret, and Evelyne Viegas (Eds.). 2009. *Proceedings of the Web Search Click Data Workshop at WSDM 2009*.

[4] Jeffrey Dean. 2009. Challenges in building large-scale information retrieval systems: invited talk. In *WSDM*.

[5] Constantinos Dimopoulos, Sergey Nepomnyachiy, and Torsten Suel. 2013. Optimizing Top-k Document Retrieval Strategies for Block-max Indexes. In *WSDM*. 113–122.

[6] Shuai Ding and Torsten Suel. 2011. Faster top-k document retrieval using block-max indexes. In *SIGIR*. 993–1002.

[7] Hui Fang, Tao Tao, and ChengXiang Zhai. 2004. A Formal Study of Information Retrieval Heuristics. In *SIGIR*. 49–56.

[8] Craig Macdonald, Iadh Ounis, and Nicola Tonellotto. 2011. Upper-bound Approximations for Dynamic Pruning. *ACM Trans. Inf. Syst.* 29, 4 (2011), 17:1–17:28.

[9] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Christina Lioma. 2006. Terrier: A High Performance and Scalable IR Platform. In *OSIR*.

[10] Matthias Petri, J. Shane Culpepper, and Alistair Moffat. 2013. Exploring the Magic of WAND. In *ADCS*. 58–65.

[11] Sebastiano Vigna. 2013. Quasi-succinct indices. In *WSDM*. 83–92.

[12] Lidan Wang, Jimmy Lin, and Donald Metzler. 2010. Learning to Efficiently Rank. In *SIGIR*. 138–145.