Efficient and Effective Higher Order Proximity Modeling

Xiaolu Lu
RMIT University
Melbourne, Australia
xiaolu.lu@rmit.edu.au

Alistair Moffat
The University of Melbourne
Melbourne, Australia
ammoffat@unimelb.edu.au

J. Shane Culpepper
RMIT University
Melbourne, Australia
shane.culpepper@rmit.edu.au

ABSTRACT

Bag-of-words retrieval models are widely used, and provide a robust trade-off between efficiency and effectiveness. These models often make simplifying assumptions about relations between query terms, and treat term statistics independently. However, query terms are rarely independent, and previous work has repeatedly shown that term dependencies can be critical to improving the effectiveness of ranked retrieval results. Among all term-dependency models, the Markov Random Field (MRF) [Metzler and Croft, SIGIR, 2005] model has received the most attention in recent years. Despite clear effectiveness improvements, these models are not deployed in performance-critical applications because of the potentially high computational costs. As a result, bigram models are generally considered to be the best compromise between full term dependence, and term-independent models such as BM25.

Here we provide further evidence that term-dependency features not captured by bag-of-words models can reliably improve retrieval effectiveness. We also present a variation on the highly-effective MRF model that relies on a BM25-derived potential. The benefit of this approach is that it is built from feature functions which require no higher-order global statistics. We empirically show that our new model reduces retrieval costs by up to 60%, with no loss in effectiveness compared to previous approaches.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models, search process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation*

Keywords

Proximity; Experimentation; Measurement

1. INTRODUCTION

Bag-of-words models are widely used in information retrieval. Regardless of whether they are based on probabilistic ranking principles, such as BM25 [29], or drawn from a language modeling family, such as the Query Likelihood model [25], those models assume either that the query terms are independent of each other, or are conditionally independent. To compute a document score, each term contributes according to its TF score and IDF weighting, with the sum across the terms taken as the score for the document as a whole. Bag-of-words models are widely used because they can be efficiently implemented, and in most applications also attain good retrieval effectiveness.

However, a range of sophisticated proximity models that allow terms to be dependent on each other have been proposed, and have shown a significant gain in retrieval effectiveness over bag-of-words models [12, 23, 32, 9, 14, 24]. Among term-dependency models, the Markov Random Field (MRF) [23] model has received considerable attention. Effectiveness is improved by considering all possible dependencies among query terms; the tradeoff cost arises in the form of higher computational costs. Bigram models can also be used as a compromise between full dependence, and fully independent models. However, the parameter tuning process for bigram models can be non-trivial, and may not obtain the expected effectiveness even if resources such as Google's N-grams are applied.

In order to achieve high effectiveness, current MRF term dependency models require global collection statistics, counting the frequency of each possible dependency. Such global statistics are expensive to obtain, except in the trivial case of single terms. Unless large amounts of storage and preprocessing time are used, global statistics describing two or more terms can only be computed at query time; the cost of doing that then becomes a significant bottleneck when evaluating term-dependency queries. In particular, two complete iterations over the postings lists of terms must be performed, meaning that long queries, or queries containing frequent words, may give rise to unacceptable query latencies.

In this paper, we first empirically reinforce the importance of proximity features in the retrieval process, and confirm that it is crucial to combine both term-independent and term-dependent score components in order to achieve maximum effectiveness. Then, to address the "global statistics" bottleneck associated with proximity models, we propose an alternative feature function that can be used in the MRF framework. We empirically test our new variant MRF model on three TREC collections, and show its effectiveness compared to other MRF models. Finally, having demonstrated the effectiveness of the approach, we turn our attention to execution cost, and show that our mechanisms reduce per-document retrieval costs by up to 60% compared to other methods.

2. BACKGROUND

Bag-of-Words Models Many different retrieval models have been proposed over the years. Most current approaches fall in to one of two categories: (i) BM25 scoring approaches; and (ii) language

ICTIR'16 , September 12–16, 2016, Newark, DE, USA

ACM 978-1-4503-4497-5/16/09 ...$15.00.
http://dx.doi.org/10.1145/2970398.2970404.

models. Other options include mechanisms based on divergence from randomness [1]. Here we focus on the unigram models associated with BM25 and with language models, because of their importance as components of proximity models. Although they have different theoretical explanations, both assume independence or conditional independence among query terms [27, 35].

Let $Q = \{q_0, q_1, \ldots, q_{k-1}\}$ be a query. Then the score of a document D relative to Q can be formulated as an accumulation of unigram scores:

$$\textit{Score}(Q, D) = \sum_{q \in Q} \textit{Score}(q, D) \tag{1}$$

The BM25 model is based on the probabilistic ranking principle:

$$\begin{aligned} \textit{BM-Score}(Q, D) &= \sum_{q \in Q} w_q \cdot \frac{f_{q,D} \cdot (1 + k_1)}{f_{q,D} + K} \\ K &= k_1 \cdot (1 - b + \frac{b \cdot |D|}{avg(|D|)}), \end{aligned} \tag{2}$$

where w_q is the IDF weighting, $f_{q,D}$ is the TF value of term q, and k_1 and b are tunable parameters. There are several variants of the BM25 model, with one categorization based on the IDF weighting, w_q. Either a Robertson-Spärck Jones parameter $w_q^{(1)}$ [28], or a Robertson-Walker parameter $w_q^{(2)}$ [17] can be applied, and calculated as:

$$w_q^{(1)} = \log \frac{N - N_q + 0.5}{N_q + 0.5} \text{ , and } w_q^{(2)} = \log \frac{N}{N_q}, \tag{3}$$

where N is the total number of documents in the collection and N_q is the number containing term q. The $w_q^{(2)}$ version is implemented in ATIRE as recent work has shown it to be the most effective in practice. [33, 34].

Language models [25] score documents using a somewhat different approach. The widely used Language Model with Dirichlet Smoothing (LMDS) can be formulated as:

$$\textit{LM-Score}(Q, D) = \sum_{q \in Q} \log \left(\frac{\mu \cdot f_{q,\mathcal{C}} / |\mathcal{C}| + f_{q,D}}{\mu + |D|} \right), \tag{4}$$

where $|\mathcal{C}|$ is the size of the collection; $f_{q,\mathcal{C}}$ is the number of occurrences of term q in the collection; $f_{q,D}$ is the number of occurrences of q in document D (that is, the TF value), and μ is the Dirichlet smoothing parameter. As for the BM25 models, there are several variants in the LM family, using different smoothing methods, and/or based on a different assumption as to the underlying distribution of terms. We consider the LMDS approach of Equation 4 to be a good representative of language models for the purposes of our experiments, and the BM25 formulation of Equation 2 to be a representative probabilistic method. Both have been empirically shown to have reasonably good effectiveness on a wide variety of test collections, and can be efficiently implemented using a document-level inverted index [37].

In order to further improve retrieval effectiveness, the BM25 and LMDS models can be combined with information gleaned from other features, either query-independent or query-dependent, with the merging process often involving linear combinations. Query-independent features include static document scores such as PageRank and quality/spam scores; and one intuitively attractive query-dependent feature is to utilize term positions and the proximity of query terms in the document. This latter approach has received considerable attention in previous work [26, 9, 23, 6, 13, 32, 24]. While there have been empirical studies showing that the use of proximity statistics enhances the effectiveness of ranked retrieval [12, 32], there are also several counterexamples that question the value of including proximity-based features in retrieval [35, 27], partly because of the expense of computing them.

In other related work, de Kretser and Moffat [10] combine localized term scores to allow terms to reinforce each other when they appear close together. Their approach makes use of a word-level inverted index so that term positions are known, but avoids the use of global statistics. The main strength of their approach is to locate more specific retrieval zones within long documents; it also can be used as an aid when preparing a document summary or caption.

Influence of Proximity Scores The question of whether proximity features boost bag-of-words scoring models arises because unigram models already incorporate some of the same influences. For example, if two query terms both appear in a document, then the document will almost certainly be scored more highly by a bag-of-words model than if only one of them appears, and it might be that the two occurrences are only rarely separated by a large span of terms anyway. In this case, using an auxiliary proximity model might not alter the underlying document ranking in any way. Part of our work here is to demonstrate the converse – that adding proximity features to the BM25 and LMDS unigram bag-of-words models unambiguously improves effectiveness, provided that the two parts are combined carefully. In particular, let $\textit{Score}(Q, D)$ be the score of any bag-of-words model, $\textit{Score}(\mathcal{I}, D)$ be the score derived solely from the model based on the proximity features in a document ($\mathcal{I}$ will be defined shortly as a set of *intervals*), and $\Pr(D = 1)$ be the probability of a document being a relevant one. Then if proximity features are a useful addition to the bag-of-words model, we would hope to demonstrate that if $\textit{Score}(Q, D_1) = \textit{Score}(Q, D_2) \wedge \textit{Score}(\mathcal{I}, D_1) > \textit{Score}(\mathcal{I}, D_2)$, then $\Pr(D_1 = 1) > \Pr(D_2 = 1)$. We present results supporting this hypothesis in Section 4.

Modeling Term Dependencies To model term dependencies, Metzler and Croft [23] propose a framework based on a Markov Random Field (MRF), an undirected graph model, in which potential functions applied over clique sets are defined according to a specific task. For IR applications, Metzler and Croft consider three generalized graph structures to represent different independence assumptions, namely Full Independence (FI), Sequential Dependence (SD) and Full Dependence (FD). There are three different types of clique sets in these structures [21]: (i) the query dependent clique set, which only contains query nodes matching subqueries $Q' \subseteq Q$; (ii) the query-document clique set, which contains query terms in Q' and a document; and (iii) the document clique set, in which only the document is considered. The first type of clique sets are query dependent features, and can be scored using IDF weightings. Any TF-IDF ranking functions can be used as feature functions of the second clique set; and the document clique set can be modeled using document-dependent features [21]. The potential function is then built from all of the feature functions.

Following the original definition of each clique set, let $T_{Q,D} = \{\{q_0, D\}, \{q_1, D\}, \ldots \{q_{k-1}, D\}\}$. Then the FI model only considers this type of clique set, and hence yields a bag-of-words model. Further, let *n*-$O_{Q,D}$ be a set of clique sets that matches phrases containing between two and n contiguous query terms, and let *n*-$U_{Q,D}$ be the set of clique sets matching subqueries formed containing between two and n query terms, without the requirement that the terms be adjacent in D. The SD model (SDM) only considers *2*-$O_{Q,D}$ which are the bigrams in the query, while the FD model (FDM) considers $T_{Q,D}$ plus *n*-$O_{Q,D}$ plus *n*-$U_{Q,D}$. The LMDS feature function is applied over these clique sets based on a set of independence assumptions.

The INQUERY interface [8] provides operators that support these

MRF ranking options. Let $\mathcal{Q}$ be the set of all two-or-more term subsets of Q. Then the MRF model is formulated as:

$$\begin{aligned} \textit{LM-M-Score}(Q,D) =& \lambda_t \cdot \textit{LM-Score}(Q,D) + \\ & \lambda_o \cdot \sum_{Q' \in \mathcal{Q}} \textit{LM-Score}(\#ow(Q'),D) + \\ & \lambda_u \cdot \sum_{Q' \in \mathcal{Q}} \textit{LM-Score}(\#uwX(Q'),D), \end{aligned} \tag{5}$$

in which λ_t, λ_o, and λ_u are weighting parameters for unigrams, ordered phrases and unordered phrases, respectively; where $\#ow(Q')$ denotes an exact match of Q' in D; and where $\#uwX(Q')$ denotes an unordered match of Q' within distance X. In FI, only the first term is considered (equivalent to $\lambda_o = \lambda_u = 0$); and SDM only considers bigram queries, in which Q' is formed from two adjacent terms in the query. All $Q' \in \mathcal{Q}$ are used in $n\text{-}U_{Q,D}$ and $n\text{-}O_{Q,D}$.

The MRF framework captures term-dependencies explicitly, and is readily adapted to make use of different potentials. Metzler [22] initially described BM25-derived potentials as well as LMDS-derived potentials, and subsequent work found that the differences in terms of effectiveness are not significant in most cases [11]. On the other hand, previous work has not explored the impact on efficiency when using different potentials.

Bigram Models as a Surrogate Both Zhai [35] and Robertson [27] note that higher-order proximity scoring is computationally intensive, and as a compromise suggest using bigrams only. One of the most widely used non-parametric bigram models, BCTP [6], is an extension of a BM25 model originally described by Rasolofo and Savoy [26]. BCTP takes the word distance between terms q_i and q_{i+1} in the document into consideration:

$$\begin{aligned} \textit{acc}(q_i,D) &= \textit{acc}(q_i,D) + w_j^{(2)} \cdot \textit{dist}(q_i,q_j)^{-2} \\ \textit{acc}(q_j,D) &= \textit{acc}(q_j,D) + w_i^{(2)} \cdot \textit{dist}(q_i,q_j)^{-2}, \end{aligned} \tag{6}$$

where $w_i^{(2)}$ and $w_j^{(2)}$ are the IDF weightings of q_i and q_j, and $\textit{dist}(q_i,q_j)$ is the distance between the bigrams, leading to:

$$\begin{aligned} \textit{TP-Score}(Q,D) = & \sum_{q \in Q} \textit{BM-Score}(q,D) + \\ & \sum_{q \in Q} \min\{w_q^{(2)}, 1.0\} \cdot \frac{\textit{acc}(q,D) \cdot (1+k_1)}{\textit{acc}(q,D) + K} \end{aligned} \tag{7}$$

in which the score $\textit{acc}(q_i,D)$ for query term q_i is then combined with its bag-of-words BM25 score. In this model, BM25 uses the $w_q^{(2)}$ IDF weighting given in Equation 3, and k_1, K are the same parameters as in Equation 2, with each query term scored in the context of the intervals formed with other terms. Like the earlier method of de Kretser and Moffat [10], only unigram global statistics are required, together with term positions within documents. In both of these models the distance between occurrences of query terms plays a role in the scoring process. The BCTP model was also used in recent work by Broschart and Schenkel [5], who explore several hybrid indexing techniques capable of balancing query efficiency and index size.

As was noted above, SDM is obtained by considering only sequential dependencies of bigrams. Instead of using the default tuning parameters in each feature function as described by Metzler [22], it is also possible to use external resources to give each clique set a different weighting. The weighted SDM (WSDM) [14, 3, 4] yields superior performance compared to SDM, achieving similar effectiveness to FDM on some queries. BCTP tends to be less competitive when compared with a sequential dependency model (SDM) [23], especially when WSDM is used. Although a carefully tuned bigram model can be better than a high-order proximity model, Huston and Croft [14] also note a degradation in effectiveness on deep-pooled collections. If unweighted SDM is applied to the extents and compared to a higher proximity model such as FDM, the bigram model is less effective. Huston and Croft [14] also hypothesize that bigram models may be better than higher-order models for some queries, and that with the proper configuration, many-term dependency models can also be improved.

Higher-Order Proximity Consider the query "women ordained church of england" (Robust04, Query 621), and a section from a relevant document, with words annotated with their positions:

> ^{0}*England* ^{1}*is* ^{2}*set* ^{3}*to* ^{4}*have* ^{5}*women* ^{6}*priests* ^{7}*within*
> ^{8}*two* ^{9}*years,* ^{10}*following* ^{11}*a* ^{12}*close* ^{13}*vote* ^{14}*in* ^{15}*the*
> ^{16}*Church* ^{17}*of* ^{18}*England's* ^{19}*general* ^{20}*synod* ^{21}*yesterday.*
> ... ^{50}*if* ^{51}*women* ^{52}*were* ^{53}*ordained.* ^{54}*Mr* ^{55}*Gummer*
> ^{56}*had* ^{57}*suggested* ^{58}*that* ^{59}*he* ^{60}*might* ^{61}*leave* ^{62}*the*
> ^{63}*Church* ^{64}*of* ^{65}*England* ^{66}*over* ^{67}*women's* ^{68}*ordination.*

In this query, "church of england" is clearly a phrase, and using the two bigrams it contains may yield a potential gain compared to a bag-of-words only query, since the phrase occurs twice in the example document. The other two terms, "woman" and "ordained", occur both as a phrase (words 67-68), and also as a proximity, separated by the word "were" (words 51-53); and all five query terms occur within a span of 6 (words 63-68), a higher-order proximity.

Multi-term dependency models assume that all terms may have informative interactions, and as a result may better capture a user's intent. These models can be roughly categorized into three different approaches, according to the proximity statistics used: (i) the statistics of a "span" [9, 30, 31] in a document; (ii) the statistics of all subqueries of the current query in a document [23]; and (iii) the statistics of term positional information [19, 36].

The use of span (or cover) statistics was proposed by Clarke et al. [9], and examined subsequently by Song et al. [30]. A span is a minimal interval formed by occurrences of two query terms. For example, consider the subquery $Q' = \{\text{england}, \text{women}\}$ when applied to the example document. The interval $[0\ldots5]$ is optimal, whereas the interval $[0\ldots51]$ is not, since the latter is a superset of the former. A more detailed definition is provided by Clarke et al. [9]. Span statistics are straightforward to use since the optimal intervals are naturally separated, even without specifying a distance constraint. Both Clarke et al. and Song et al. consider a chain of such intervals, the difference being: (i) Song et al. consider a non-overlapping version of "covers", and apply different weights using a function parameterized on both the number of query terms captured, and the distance of an interval; and (ii) Song et al. reformulate the BM25 model using redefined term contributions. The work of Song et al. [30] is based on a similar strategy as is used in the earlier BCTP model, and query terms are scored using different contexts, where "contexts" are spans containing the query term.

Unlike other models that have a separate bag-of-words component, these models integrate the proximity contributions for each query term directly into the BM25 computation. Let $I[p_\ell \ldots p_r]$ be a span, and the query terms bounded by the span be $I \cap Q$. Then $\textit{acc}(q,D)$ is calculated for a query term contribution as:

$$\begin{aligned} \textit{Span-Score}(Q,D) &= \sum_{q \in Q} w_q^{(1)} \cdot \frac{\textit{acc}(q,D) \cdot (1+k_1)}{K + \textit{acc}(q,D)} \\ \textit{acc}(q,D) &= \sum_{q \in I} \frac{|I \cap Q|^\lambda}{(p_r - p_\ell)^\gamma}, \end{aligned} \tag{8}$$

where K is the same BM25 parameter as in Equation 2, and λ and γ are tuning parameters that are set on a per-collection basis. Song et al. show significant improvements relative to their baselines, and additional gains are made by Svore et al. [31] using features generated from external data sources.

The "window" statistics used in MRF [23] are a little different from these optimal intervals. The distance constraint must be specified, otherwise the computation degrades into a ranked Boolean query [8]. Consider a subquery $Q' = \{\text{england}, \text{ordain}\}$ for the same example document. If there is no distance constraint, the first match is $[0\ldots25]$, but if the default Indri settings are used, which is $4 \cdot |Q'|$ [23], the result will then be $[18\ldots25]$. As discussed above, a matched interval is an instance of a clique defined in the MRF.

The FDM approach considers all possible dependencies for all query terms. To make sure all of the clique sets are considered, the model scores $2^{|Q|} - |Q| - 1$ unordered window subqueries, and $|Q| \cdot (|Q| - 1)/2$ ordered window subqueries. Consider the feature function of an unordered window match as an example. If *#uw*(Q') is the set of matched intervals of a subquery Q', then the feature function is defined as:

$$\textit{LM-Score}(\textit{\#uw}(Q'), D) = \log\left(\frac{\mu \cdot f_{uw,\mathcal{C}}/|\mathcal{C}| + f_{uw,D}}{\mu + |D|}\right), \quad (9)$$

where $f_{uw,\mathcal{C}}$ is the number of matching unordered windows in the whole collection, and $f_{uw,D}$ is the within-document frequency of the unordered window. Other window statistics are calculated similarly, using the LMDS scoring regime as a feature function. Once all components have been computed, they are combined.

This model is still considered to be one of the best known higher-ordered proximity models, and the experimental results of Huston and Croft [14] on the TREC8 Robust04 Task and on GOV2 show that significant gains are possible compared to lower-order computations. Although a limited window size is applied in the scoring process, for simplicity this form of FDM actually omits the differences between cliques in the same clique set. Whilst effective, this model is also expensive to compute for two reasons: (i) a large number of subqueries must be enumerated (exponential in $|Q|$); and (ii) the need for global proximity statistics mandates a two-pass retrieval process. Macdonald et al. [20] observe that one way to avoid computing the global statistics is to use a constant, as is done in the Ivory System[1]. However, it is unclear how best to tune this value, and it may be sensitive to the collection being used. An alternative is to precompute and store the global statistics at indexing time, but enumerating all possible subqueries is still costly. Huston et al. [15] show that even considering only N-grams can be expensive. A possible trade-off is to consider an approximated scoring mechanism instead of an exact one, as is done by Elsayed et al. [11]. However, Elsayed et al. only consider the SDM model, in which only subqueries with adjacent term pairs are considered.

Positional Models The final class of models discussed here utilize the position information of terms in a document [19, 36]. These models focus on finding propagation functions to model position information of terms. For example, the Positional Language Model [19] generates all possible position propagations of the query terms; a computation that is costly, especially for long queries and large collections. Huston and Croft [14] give a detailed comparison of positional models; their results suggest that, although WSDM is best when averaged across all queries tested, the FDM model could still be improved with more time and effort.

Passage Retrieval Instead of applying the ranking mechanism at the document level, it is also possible to partition the document into fixed-length overlapping or non-overlapping passages, and score these smaller pieces. Either an entire document, or a single passage can then be returned to the user [16]. Window-based methods can be implemented and represented using proximity operators [7], but differ from what we describe here. First, a fixed length passage can be generated dynamically by identifying a starting point based on one of the query terms, and computing a similarity score over the remainder of the extent, which does not necessarily contain all of the query terms; in contrast, the proximity features described in this work are extracted using a window capturing each subquery. And second, proximity-based models of the kind we consider here score at the document level, while passage retrieval uses passage-level statistics, including, for example, passage-level TF values.

[1] `http://lintool.github.io/Ivory/index.html`

3. A NEW APPROACH

We now describe a new form of high-order proximity computation that also does not require global statistics.

Intervals We use the definition of intervals originally proposed by Clarke et al. [9], but employ non-overlapping intervals as suggested by Song et al. [30]. That is, the next candidate interval of the current subquery must start at a position beyond the right end of the previous optimal interval. We also model term dependencies using the MRF framework, so that non-overlapping optimal intervals are still instances of the corresponding cliques. To determine the dependencies, we enumerate all possible matches for all subqueries, as is done for FDM, and the same span can be repeatedly used and scored differently according to the matching subquery, rather than just being included once.

Consider the sample document again, and the two subqueries: $Q_0 = \{\text{woman}, \text{england}\}$ and $Q_1 = \{\text{woman}, \text{ordained}, \text{england}\}$. According to the interval definition, the two sets of intervals are:

$$\mathcal{I}(Q_0) = \{[0\ldots5],\ [18\ldots51],\ [65\ldots67]\} \text{ and}$$
$$\mathcal{I}(Q_1) = \{[18\ldots53],\ [65\ldots68]\}.$$

The two sets may contain overlapping intervals. In the example, $[65\ldots67] \in \mathcal{I}(Q_0)$ is dominated by the interval $[65\ldots68] \in \mathcal{I}(Q_1)$. In the MRF framework, the smaller interval is not dropped, since it arises from a different subquery. The interval-finding process is iterated until the positions for all query terms are exhausted.

Scoring Intervals Each extracted interval is then scored using a feature function. In our case, we define the function in such a way that an interval will be scored more highly if it:

- captures more query terms over a shorter distance;
- captures more important terms; and
- is not bounded by common terms.

The first of these relationships is easiest to motivate, and has been widely applied in previous proximity ranking models. For example, in MRF models, only windows of up to some maximum length are considered. The second relationship is used to distinguish different subqueries. A complex weighting assignment process can be used to implement this effect, such as training with external resources, the approach used in WSDM. We prefer a more straightforward option, and use the IDF weightings of the terms. The third relationship relates to the interval instances that match a single subquery. In this case, we use the boundary term to determine whether the current matching is an imprecise representation of the current subquery, because when the LHS or RHS term is a common one, the interval is less representative than ones bounded by less-common terms. As a concrete example, consider the interval statistics for another subquery: $\mathcal{I}(\text{woman}, \text{of}, \text{england}) =$

$\{[0 \ldots 17], [18 \ldots 64]\}$. Both intervals are right bounded by the term "of", which is a common term. Although both of these intervals are valid candidates for this subquery, if the document also contained a valid interval such as "woman of england" that had the "of" in the middle, then all other things being equal, that third one should be favored. Indeed, intervals in $\mathcal{I}(\text{woman, of, england})$ have much the same effect as intervals in $\mathcal{I}(\text{woman, england})$ when "of" is one of the two endpoints.

To embody these three relationships, we suggest that candidate interval $I[p_\ell \ldots p_r]$ of a subquery Q' be scored as:

$$\mathit{Score}(I, D) = w_\ell^{(2)} \cdot w_r^{(2)} \cdot (p_r - p_\ell + 1)^{-2} \qquad (10)$$

where $w_\ell^{(2)}$ and $w_r^{(2)}$ are the IDF weights of the interval's two boundary terms, p_ℓ and p_r are the positions of those boundaries, and $w_q^{(2)}$ is the IDF weight of a term that is within the interval. The $w_r \cdot w_\ell$ component discounts intervals that are bounded by common terms (the third relationship), and borrowing from Büttcher et al. [6], we use $(p_r - p_\ell + 1)^{-2}$ to discount for distance. In addition, rather than count the number of query terms in the subquery, we use $\min\{w_q, 1.0\}$ to down-weight subqueries that consist of low weighted terms, thereby also incorporating the second relationship; an adjustment that is motivated at the level of intervals, but for convenience is included as a component of Equation 11. This factor is computed automatically during query processing, and does not require additional parameter tuning.

Combining Interval Scores Trotman et al. [34] observe that while there is no single ranking function that is consistently better than the others, the BM25 variants generally achieve good performance. Let $\mathit{Score}(Q', D)$ be the score of set of intervals $\mathcal{I}$ that match a subquery Q'. For our purposes the score for those intervals is defined in a BM25-like way as:

$$\mathit{Score}(Q', D) = \frac{\sum_{I \in \mathcal{I}} \mathit{Score}(I, D) \cdot (1 + k_1)}{\sum_{I \in \mathcal{I}} \mathit{Score}(I, D) + K'}$$
$$K' = K \cdot \left(\sum_{q \in Q'} \min\{w_q^{(2)}, 1.0\} \right)^2 , \qquad (11)$$

where K is the BM25 parameter used in Equation 2. As noted earlier, one of the reasons that FDM is expensive is because it uses collection-based frequencies, and hence two passes through the index. To avoid that cost, we refrain from any use of global higher-order counts in Equation 11.

We also include ordered phrase relationships in our overall formulation, even though phrases can sometimes degrade retrieval effectiveness. Let $\mathcal{Q}$ be the subquery set, and $\mathcal{Q}' \subset \mathcal{Q}$ be the set of sequentially dependent subqueries, that is, subsets of the query Q containing two or more terms in which term order is preserved relative to Q. We combine all of these features in the usual weighted manner, using the BM25 derived potentials in MRF, and obtain as a result our proposed method:

$$\begin{aligned} \mathit{Score}(Q, D) = (1 - \lambda) \cdot \mathit{BM\text{-}Score}(Q, D) \\ + \lambda \cdot \sum_{Q' \in \mathcal{Q}'} \mathit{Score}(Q', D) \\ + \lambda \cdot \sum_{Q' \in \mathcal{Q}} \mathit{Score}(Q', D) , \end{aligned} \qquad (12)$$

that is, $(1-\lambda)$ times the FI score, plus λ times the combined phrase score (over $Q' \in \mathcal{Q}'$) plus proximity score (also over $Q' \in \mathcal{Q}$).

Equation 12 makes it clear that Lkp – the name we give to this approach – captures both independent relationships as well as sequential and full dependencies among query terms, both of which may improve query effectiveness; and in doing so, echoes the factors employed in FDM. In this initial model the proximity statistics required will still grow rapidly relative to the query length, and lead to significant cost. However, global weightings are only required for unigram terms, which makes this model more amenable to further efficiency optimizations, since individual term weightings are stored in the inverted index. Moreover, our model benefits from optimal intervals, and, as described, does not require that a distance constraint be added as a further parameter to be tuned and set.

Collection	N	$\lvert\mathcal{C}\rvert$	Queries	
			T-query	U-query
TREC8	0.5 M	253.4 M	250	1,933
GOV2	25.2 M	23,451.8 M	150	1,496
ClueWeb09B	50.1 M	40,416.4 M	3×50	–

Table 1: Test collections used, where N is the number of documents, and $|\mathcal{C}|$ is the total number of terms. The two "Queries" columns list the number of queries for each collection, with "T" denoting TREC title queries, and "U" denoting user queries [2].

Variants Interval lengths can also be restricted if desired, with no impact on efficiency. Moreover, since our model is a variant of the original MRF model, if only bigram subqueries are considered, it reduces to a bigram model. Hence, three versions are proposed:

- Lkp: as described by Equations 10, 11, and 12;
- L$k$fp: as for Lkp, but with intervals restricted to a maximum distance, set as $4 \cdot |Q'|$, as for FDM; and
- L2p: as for Lkp, but considering bigrams only, that is, pairs of words that are adjacent in the query.

4. EXPERIMENTS

We now describe the results of experiments on the effectiveness and efficiency of these three new similarity formulations.

Experimental Setup We use the TREC8, GOV2, and ClueWeb09-Category-B collections in our experiments, always in non-stopped forms, since proximity and phrase-based models can use combinations of stopwords to improve overall effectiveness for some queries (consider the query "the who" for example). Indexing is carried out using Indri[2], with a Krovetz stemmer. Two broad types of query sets are used. The primary resources are the title queries of the Robust Task 2004 (R04, with 250 queries in total), the Terabyte Task from 2004 to 2006 (TB04–TB06, with 150 queries in total), and the three ClueWeb Adhoc Tasks from 2010 to 2012 (C10–C12, with 50 queries each). As a secondary resource, we augment these with the user-generated query variants collected by Bailey et al. [2] in connection with the Robust03 and Terabyte04 Tasks, denoted here as R03-U and TB04-U. Table 1 summarizes the three document collections and seven query sets used; and Figure 1 summarizes the lengths of the five TREC-title query sets.

The top 1,000 documents are identified for each query in all effectiveness measurements. The TREC8 and GOV2 results are reported using two relatively deep metrics, average precision (AP) and rank-biased precision (RBP@0.95), in both cases based the full run and all available judgments. The three ClueWeb collections have shallow judgments, and hence are scored using expected reciprocal rank to depth 20 (ERR@20) and a more top-weighted

[2] http://www.lemurproject.org/indri/

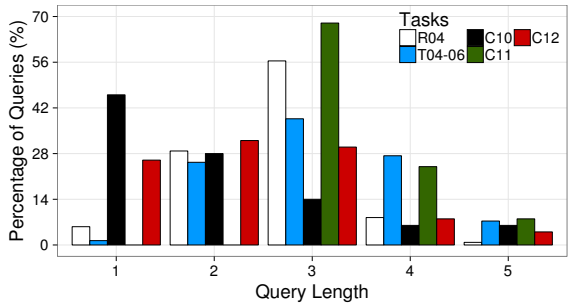

Figure 1: TREC title query lengths for five query sets: Robust04 (a subset of Topics 301–700), Terabyte04–06 (Topics 701–850), ClueWeb 2010 (Topics 51–100), ClueWeb 2011 (Topics 101–150), and ClueWeb 2012 (Topics 151–200).

Models	Type	Global lvl.	Params.
BM25$^{(1)}$	BM25	term	$k_1 = 0.9,\ b = 0.4$
BM25$^{(2)}$	BM25	term	$k_1 = 0.9,\ b = 0.3$
LMDS	LM	term	$\mu = 2500$
SDM	LM-B	bigram	$\lambda_o = 0.15,\ \lambda_u = 0.05$
FDM	LM-N	n-gram	$\lambda_o = 0.1,\ \lambda_u = 0.1$
BCTP	BM25-B	term	–
L2p	BM25-B	term	$\lambda = 0.4$
Lkp	BM25-N	term	$\lambda = 0.4$
L$k$fp	BM25-N	term	$\lambda = 0.4$

Table 2: Models used in experiments. "Type" indicates the family origin of the method, with suffix "B" denoting a bigram model, and "N" a higher-order proximity model. Column "Global lvl" indicates the highest level that global statistics are used; and "Params" lists the parameters, with bag-of-words parameters re-used in the proximity models and not listed a second time. For FDM and SDM we use the recommended configurations in the original paper, and for BM25 we use the settings recommended by Trotman et al. [34]. The parameters for the three new models were trained using the Robust04 query set and the TREC8 collection.

rank-biased precision (RBP@0.8), again based on the full run and all available judgments. Of these three metrics, ERR@20 is computed using graded relevance, with the maximum gain set according to the corresponding task; RBP and AP are both computed using binarized judgments. Experiments were performed on an Intel Xeon E5 CPU, 256 GB RAM, and RHEL-v6.3 Linux, implemented in C++, and compiled using GCC 4.8.1 with –O2 optimization.

Baselines The TREC title queries are used as effectiveness baselines. We use BM25 with Robertson-Walker $w^{(2)}$ IDF weighting as the baseline FI model, and also list the effectiveness scores for Indri's built-in version of Okapi BM25, which uses Robertson-Spärck Jones IDF weighting $w^{(1)}$. Methods from the language modeling family are also included; these results are generated using Indri and the default configurations. To validate the experiments of Huston and Croft [14], we also include the BCTP bigram model as a baseline. Parameters for these reference systems are listed in Table 2.

New Proximity Methods The Lkp method described in Equations 10, 11, and 12 is included in Table 2, as are the L$k$fp variant that employs a distance constraint (set to match the MRF models, that is, $4 \cdot |Q'|$), and the L2p method, which makes use of bigrams

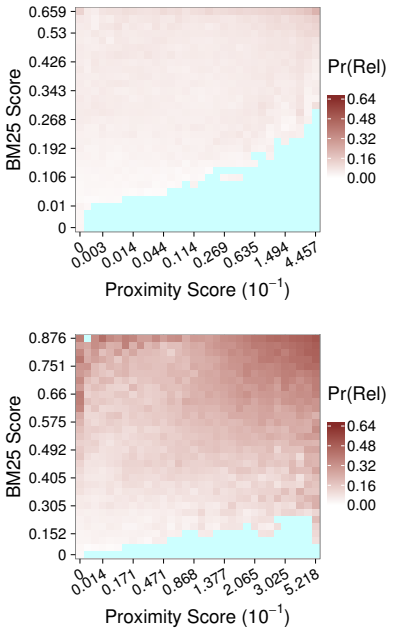

Figure 2: The relationship between BM25$^{(2)}$ scores and proximity scores. The top plot uses the R03 queries and the judged documents from the TREC8 collection; the lower plot the TB04 queries and the GOV2 collection. TREC title queries were used as well as the user queries. Measured document relevance in each of $1024 = 32^2$ bins is plotted, with blue cells having insufficient data to form estimates.

only. Parameters for the three new methods were trained using the Robust04 queries on TREC8 collection, and then applied to the other collections without further adjustment.

Evidence for Proximity Our first experiment tests the hypothesis stated in Section 2: that the additional proximity features included in Lkp and L$k$fp are not captured by the simpler bag-of-words models. Taking *P-Score*(Q, D) to be the measured proximity score, and *FI-Score*(Q, D) as the score of any bag-of-words model, we wish to know, for two documents D_0 and D_1 that have (approximately) equal scores according to *FI-Score*(), if there is a residual effect that reveals a correlation between their *P-Score*() scores and the probability of being relevant.

We use the TREC title queries with more than one query term, together with R03-U and TB04-U in this experiment. Queries are treated as being independent even if derived from the same topic. On average, there are 1,512 judged documents per topic for R03, of which an average of 78.15 are relevant. That is, across the judged documents the average probability of relevance is 5.04%. For the GOV2 TB04 task there are an average of 1,185 judged documents per topic and 216.7 relevant, giving a background relevance probability of 18.73%. The experimental process carried out was:

(1) For each of the queries Q in the query set, and each judged document D, calculate *BM-Score*(Q, D) and *P-Score*(Q, D).

(2) Divide each of *P-Score*(Q, D) and *BM-Score*(Q, D) into 32 equal ranges after normalizing the score ranges to $[0, 1]$ on a per-query basis, giving 1024 bins in total.

(3) For each bin $B_{i,j}$, estimate $\Pr(D = 1 \mid D \in B_{i,j})$ using the relevance judgment associated with each document placed in to

	Robust04		GOV2 (TB04–06)	
	AP	RBP@0.95	AP	RBP@0.95
BM25$^{(1)}$	0.254	0.318	0.293	0.489
BM25$^{(2)}$	0.256	0.320	0.294	0.491
LMDS	0.248	0.308	0.290	0.474
SDM	0.262	0.322	0.319	0.506
FDM	0.264	0.322	**0.325**	0.512
BCTP	0.262	0.324	0.312	0.514
L2p	0.267	0.330†	0.320	**0.517**
Lkp	0.269	0.331†	0.321	0.515
L$k$fp	**0.270**†	**0.332**†	0.324	0.516
BestTrecRun	0.359	0.397	0.407	0.595

Table 3: Effectiveness of models on TREC8 and GOV2, using two metrics. Statistical tests were performed, with † indicating significance relative to FDM at $p = 0.05$. Note that while the TREC8/R04 combination was also used as training to set the parameter λ used in Lkp, L$k$fp, and L2p (Table 2), the generated scores are relatively insensitive to the value used.

that bin by any of the queries, possibly counting each document multiple times.

Results for the BM25$^{(2)}$ model are plotted in Figure 2, with blue cells indicating bins with fewer than 100 documents, for which it would be misleading to estimate relevance probabilities; and with dark cells indicating regions of high probability. Relevant documents receive zero scores if none of the query terms appear in them, making the first bin bigger than the others.

As expected, the higher the BM25$^{(2)}$ score, the higher the probability of relevance, and likewise for proximity scores. (These two claims can be confirmed by projecting the bin values to the vertical and horizontal axes respectively; in the interests of space we do not show these additional graphs.) If either of these factors were sufficient in their own right to predict relevance, with no influence from the other, the patterns of shade in the graphs would be parallel to the corresponding axis. In fact, the darkest area in both of the heatmaps is located in the upper-right corner, suggesting that a combination of BM25$^{(2)}$ and proximity score is a stronger indication of relevance than either of them is alone. We also carried out the same process using LMDS as the bag-of-words model, with a similar pattern of results. Note that the lighter shades for R03 compared to TB04 are due to the overall lower background probability.

Effectiveness on TREC8 and GOV2 Table 3 gives effectiveness scores for the models listed in Table 2, using the TREC8 and GOV2 data, and queries R04 and TB04–TB06 respectively, with no stopping, and scoring based on runs of 1,000 returned documents. No RBP residuals (the score range allocated to unjudged documents) are shown in this table; but across the experiments listed, they were consistently around 0.03, because of the deep judgments available for these collections. The new L2p and Lkp methods are both better than the baseline BM25$^{(2)}$ model; the difference is significant at $p < 0.05$ when a two tailed t-test is performed, for both the AP and RBP@0.95 metrics. The higher-order proximity models also have a slight advantage compared to the bigram models, a relationship that echoes that between SDM and FDM. When adding the additional distance constraint on Lkp to compute L$k$fp, observed effectiveness again increases very slightly. No training was undertaken in this regard, and further exploration may lead to a heuristic that obtains additional improvement. Huston and Croft [14] report an AP score of 0.329 on GOV2 for their WSDM-Int method, suggesting that applying additional resources to reweight subqueries may further improve performance.

Proximity improves performance on some queries and degrades it on others. The top row of graphs in Figure 3 plots the percentage change of score for Lkp compared to BM25$^{(2)}$, with queries grouped by the extent of the change in AP score measured on that query. All four of the methods plotted generate more queries with gains than there are queries with losses, and hence give rise to overall improved performance. The three graphs in the lower half of Figure 3 capture a similar comparison, but this time relative to the SDM scores. The situation with regard to improvement is less clear cut for BCTP in the first two panes, reflecting the overall rates listed in Table 3, but in the third pane, for the C10–12, all four of the methods plotted gain an advantage.

Despite the overall benefit of proximity-based methods, up to a third of queries give worse effectiveness using Lkp than they do using the BM25$^{(2)}$ bag-of-words approach. Table 4 lists the eight most degraded queries when BM25$^{(2)}$ is compared to Lkp, with degradation measured by percentage loss in AP. While there is no obvious common element to these queries, if one could be identified based on index-resident information (such as IDF values), then a hybrid system that "dialed down" the value of λ (Equation 12) for certain queries could, potentially, be better than both BM25$^{(2)}$ and Lkp. This is an area for future investigation.

Effectiveness on ClueWeb09B Table 5 lists retrieval effectiveness outcomes using the three ClueWeb query sets C10, C11, and C12; and the two rightmost plots in Figure 3 show the query improvement breakdown relative to BM25$^{(2)}$ and to SDM, aggregated across the three same three query sets. The BM25 baseline performs very well on the first of the three set of queries when measured using ERR, and BCTP works well on the other two sets when measured using RBP. Part of the explanation for the difference compared to the R04 and TB04–06 outcomes is that these queries contain more common words; the ClueWeb queries are also shorter overall, as is illustrated in Figure 1. Note also the relatively high RBP residuals observed for C12 in particular; that these are comparable in magnitude with the actual RBP scores is a warning that interpretation of these results needs to be undertaken with caution.

Table 6 lists the most badly affected queries when BM25$^{(2)}$ is compared to Lkp, now with "affected" determined by the percentage decrease in ERR. As with Table 4, there is no simple explanation as to why these queries suffer from the inclusion of proximity factors in the scoring regime. Note also – again, in common with Table 4 – that the FDM approach handles these queries just as poorly, and that the increased RBP residuals indicate that some of the loss of effectiveness may be a consequence of increased numbers of unjudged documents being retrieved compared to the BM25$^{(2)}$ runs. Were these documents to be judged, some of them might be relevant, lifting the Lkp scores.

Retrieval Cost of Proximity Models Table 7 lists per-document retrieval times. To compute these numbers, the time taken to compute the top 1,000 answers for a query was measured, and then divided by the number of documents scored in order to obtain a per-document cost. In all of these methods the number of documents scored is taken to be the union of the query terms' postings lists. Repeating that process across all of the queries in a collection allowed median, mean, and maximum values to be computed. Intervals for the L2p, Lkp, and L$k$fp methods were computed using the method described by Lu et al. [18].

The BM25 implementation is the fastest – it works solely with document-level statistics, and doesn't access term positional infor-

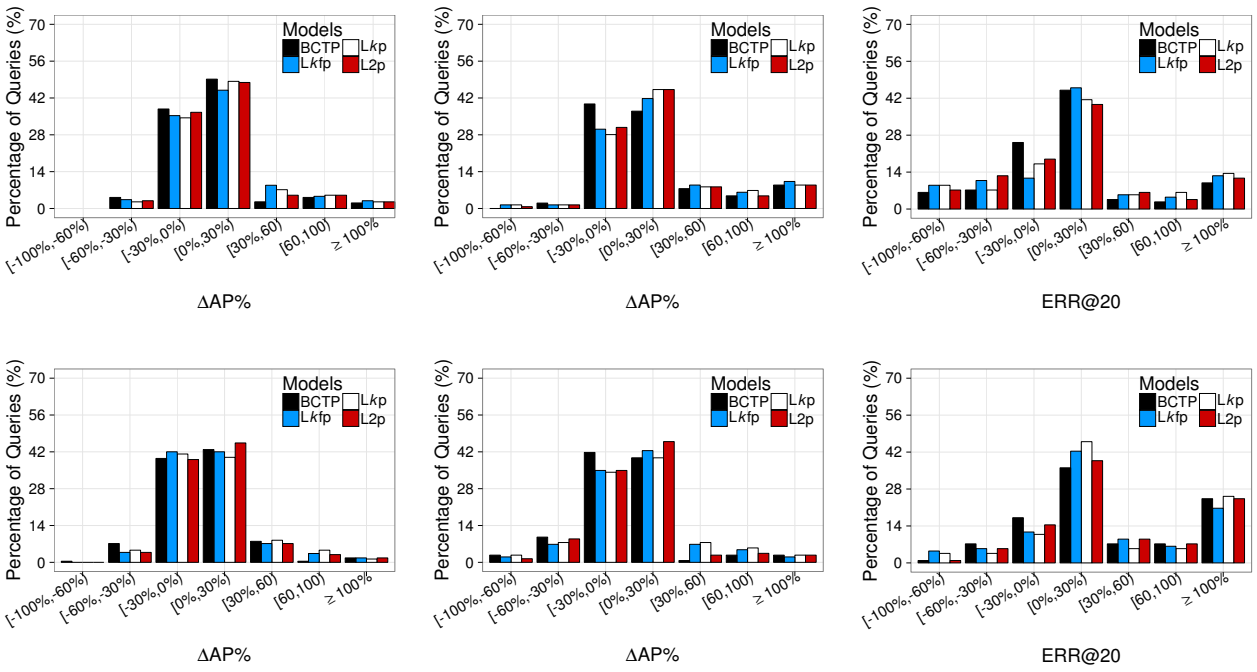

Figure 3: The first row shows the number of queries that fall into each of a set of percentage-difference buckets for each of Lkp, L$k$fp, L2p and BCTP, relative to the BM25 baseline run; and the second row shows the same computation carried out relative to SDM. The two leftmost plots are for R04 measured using AP; the two middle plots are for TB04 also measured using AP; and the two right-most plots are for C10-12 measured using ERR@20. On all three collections, we only consider queries with more than one term. The total number of queries used in the comparisons are 238, 147, and 112, respectively across the three columns.

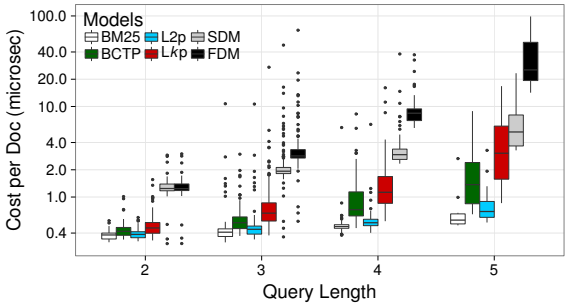

Figure 4: Per-document retrieval costs in microseconds, grouped by query length, with a logarithmic scale on the vertical axis. The measured query processing times are aggregated across all of the queries of lengths 2 to 5 in TREC title queries: R04, TB04–06, and C10–12. Unstopped indexes are used, and no query stopping.

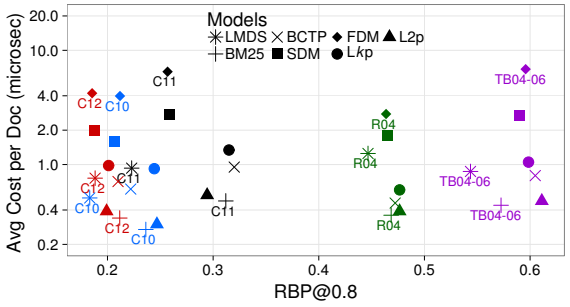

Figure 5: Trade-off graph for all retrieval models, across all collections, as indicated by the different colors. Time is measured as the mean cost per document scored in microseconds, and effectiveness using RBP@0.8.

mation. The L2p method is close behind, with times that are better than or the same as LMDS, and for the most part better than SDM. The Lkp approach is third overall in terms of speed, and it also executes more quickly than does SDM. As expected, FDM is slow – the cost of gathering the term statistics in a first pass is a substantial burden. The high cost of computing large numbers of dependencies on long queries shows in the very high maximum times associated with FDM, and to a lesser extent, Lkp. Note that the measurement methodology favors SDM and FDM, since every document in which any query term appears is assumed to be scored, amortizing the cost of the first processing pass over a large number of evaluations. A more likely usage scenario would be to compute proximity statistics during a second phase, re-scoring a small subset of the documents that have been extracted using an efficient early stage candidate identification process. In this scenario the per-document costs shown in Table 7 remain accurate for the new methods, but are under-estimates for the SDM and FDM approaches. Dynamic pruning processes such as WAND, which reduce the number of documents scored, will have a similar effect.

Two further graphs conclude our presentation. Figure 4 plots per-document querying costs as a function of query length, and shows that SDM and FDM costs grow dramatically on long queries, and that L2p is only a little slower than BM25, at all query lengths.

Qid	Query	BM25$^{(2)}$		Lkp		Lkp	FDM
		AP	RBP@0.95	AP	RBP@0.95	ΔAP%	ΔAP%
TB04-749	puerto rico state	0.165	0.451 + 0.05	0.037	0.070 + 0.01	−78%	−54%
TB04-720	federal welfare reform	0.266	0.582 + 0.05	0.080	0.223 + 0.01	−70%	−46%
R04-628	us invasion of panama	0.249	0.277 + 0.00	0.109	0.124 + 0.00	−56%	−39%
R04-381	alternative medicine	0.058	0.102 + 0.00	0.027	0.083 + 0.00	−54%	−69%
TB06-832	labor union activity	0.159	0.581 + 0.02	0.084	0.431 + 0.19	−47%	−17%
R04-389	illegal technology transfer	0.026	0.209 + 0.00	0.014	0.156 + 0.02	−45%	−50%
R04-437	deregulation gas electric	0.003	0.018 + 0.02	0.002	0.002 + 0.06	−41%	−8%
TB05-799	animals in alzheimer s research	0.312	0.352 + 0.01	0.184	0.324 + 0.05	−41%	−68%

Table 4: The eight queries for which Lkp suffers the greatest degradation compared to BM25$^{(2)}$ on the R04 and TB04–06 query sets, measured using percentage difference in AP score. The final column shows the degradation of FDM relative to BM25$^{(2)}$.

	C10		C11		C12	
	ERR@20	RBP@0.8	ERR@20	RBP@0.8	ERR@20	RBP@0.8
BM25$^{(1)}$	0.112	0.230 + 0.10	0.130	0.241 + 0.13	0.125	0.200 + 0.18
BM25$^{(2)}$	**0.121**	0.236 + 0.02	0.166	0.312 + 0.02†	0.144	**0.212 + 0.12**
LMDS	0.096	0.183 + 0.03	0.109	0.223 + 0.05	0.128	0.188 + 0.14
SDM	0.093	0.207 + 0.02	0.145	0.258 + 0.03	0.113	0.188 + 0.12
FDM	0.096	0.212 + 0.03	0.147	0.257 + 0.03	0.130	0.188 + 0.12
BCTP	0.105	0.222 + 0.02	0.162	**0.320 + 0.02**†	0.155	0.210 + 0.10
L2p	0.117†	**0.247 + 0.03**†	0.150	0.294 + 0.02	0.157	0.199 + 0.12
Lkp	0.114†	0.244 + 0.04	**0.173**	0.315 + 0.04†	0.164	0.201 + 0.13
L$k$fp	0.114†	0.237 + 0.03	0.158	0.308 + 0.05†	**0.166**	0.200 + 0.13
BestTrecRun	0.226	0.415 + 0.00	0.223	0.361 + 0.00	0.290	0.362 + 0.00

Table 5: Evaluation results of different models on ClueWeb09B, using queries C10, C11, and C12, scored using ERR to depth 20 and RBP@0.8 using all available information. Statistical tests were performed, with † indicating significance relative to FDM at $p = 0.05$.

Finally, Figure 5 plots methods/collections in a trade-off graph in which the relationship between retrieval effectiveness and retrieval cost can be seen clearly. Each collection is colored differently, and each shape corresponds to a retrieval model. The benefits of our Lkp (the solid circles) and L2p (the solid triangles) methods then become apparent – they provide high levels of retrieval effectiveness at speeds comparable to those attained by the bag-of-words BM25 mechanism.

5. CONCLUSION

We have described a new way of incorporating proximity features into a BM25-like retrieval mechanism. The result compares favorably with FDM in terms of retrieval effectiveness, but executes substantially faster. In addition, it does not require global statistics, and hence can be applied in any pass of the retrieval process, without becoming proportionately more expensive. When the same scoring computation is restricted to query bigrams, speed is close to that of the simpler bag-of-words BM25 approach. It is worth noting that additional efficiency can be achieved by indexing some or all higher order term dependencies. This is an interesting problem in its own right, and several prior studies have proposed time-space trade-offs. Our current approach employs no additional space beyond the inclusion of positional information; how to strategically employ further index space is an area for future work.

Acknowledgment This work was supported by the Australian Research Council's *Discovery Projects* Scheme (DP140101587 and DP140103256). Shane Culpepper is the recipient of an Australian Research Council DECRA Research Fellowship (DE140100275).

References

[1] G. Amati. *Probability models for information retrieval based on divergence from randomness*. PhD thesis, University of Glasgow, 2003.

[2] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. User variability and IR system evaluation. In *Proc. SIGIR*, pages 625–634, 2015.

[3] M. Bendersky and W. B. Croft. Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proc. SIGIR*, pages 941–950, 2012.

[4] M. Bendersky, D. Metzler, and W. B. Croft. Parameterized concept weighting in verbose queries. In *Proc. SIGIR*, pages 605–614, 2011.

[5] A. Broschart and R. Schenkel. High-performance processing of text queries with tunable pruned term and term pair indexes. *ACM Trans. Inf. Sys.*, 30(1):1–32, 2012.

[6] S. Büttcher, C. L. A. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proc. SIGIR*, pages 621–622, 2006.

[7] J. P. Callan. Passage-level evidence in document retrieval. In *Proc. SIGIR*, pages 302–310, 1994.

[8] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY retrieval system. In *Proc. DEXA*, pages 78–83, 1992.

[9] C. L. A. Clarke, G. V. Cormack, and E. A. Tudhope. Relevance ranking for one to three term queries. *Inf. Proc. & Man.*, 36(2): 291–311, 2000.

[10] O. de Kretser and A. Moffat. Effective document presentation with a locality-based similarity heuristic. In *Proc. SIGIR*, pages 113–120, 1999.

[11] T. Elsayed, J. Lin, and D. Metzler. When close enough is good enough: Approximate positional indexes for efficient ranked retrieval. In *Proc. CIKM*, pages 1993–1996, 2011.

Qid	Query	BM25$^{(2)}$		Lkp		Lkp	FDM
		ERR@20	RBP@0.8	ERR@20	RBP@0.8	ΔERR@20%	ΔERR@20%
C11-150	tn highway patrol	0.034	0.022 + 0.00	0.000	0.000 + 0.00	−100%	−100%
C12-161	furniture for small spaces	0.003	0.004 + 0.06	0.000	0.000 + 0.26	−100%	−100%
C12-169	battles in the civil war	0.069	0.044 + 0.05	0.000	0.001 + 0.26	−100%	−86%
C11-120	tv on computer	0.032	0.076 + 0.31	0.000	0.000 + 0.62	−100%	−74%
C12-184	civil right movement	0.165	0.188 + 0.50	0.000	0.000 + 0.68	−100%	−25%
C11-115	pacific northwest laboratory	0.063	0.160 + 0.00	0.010	0.017 + 0.08	−83%	−80%
C11-106	universal animal cuts reviews	0.048	0.134 + 0.00	0.010	0.015 + 0.02	−80%	−68%
C11-145	vines for shade	0.239	0.405 + 0.00	0.052	0.078 + 0.02	−78%	−83%

Table 6: The eight queries for which Lkp suffers the greatest degradation compared to BM25$^{(2)}$ on the C10, C11, and C12 query sets, measured using percentage difference in ERR score. The final column shows the degradation of FDM relative to BM25$^{(2)}$.

	R04			TB04–TB06			C10			C11			C12		
	Med.	Avg.	Max.	Med.	Avg.	Max.	Med.	Avg.	Max.	Med.	Avg.	Max.	Med.	Avg.	Max.
BM25	**0.37**	**0.36**	**0.53**	**0.44**	**0.44**	**0.67**	**0.40**	**0.26**	**0.67**	**0.46**	**0.48**	**0.65**	**0.42**	**0.34**	**0.60**
LMDS	0.78	1.25	1.23	0.85	0.87	1.36	0.67	0.48	1.35	0.86	0.93	1.25	0.70	0.76	6.97
BCTP	0.43	0.46	1.42	0.60	0.80	6.90	0.43	0.61	8.89	0.66	0.95	8.27	0.48	0.71	7.24
SDM	1.81	1.81	9.63	2.37	2.68	9.12	1.17	1.58	16.65	2.24	2.76	14.69	1.39	1.99	14.16
FDM	2.67	2.77	15.48	3.76	6.84	74.84	1.18	3.98	73.28	3.43	6.50	32.59	1.42	4.21	59.60
L2p	0.39	0.39	0.63	0.47	0.48	0.95	0.43	0.30	1.32	0.50	0.54	1.24	0.45	0.39	1.17
Lkp	0.54	0.60	2.25	0.80	1.05	9.31	0.45	0.92	12.95	0.88	1.34	11.17	0.68	0.98	10.20

Table 7: Retrieval cost in microseconds per document scored, using non-stopped indexes. In the C10 results, Query 70 is dropped because of its extreme computation cost ("to be or not to be").

[12] D. Hawking and P. Thistlewaite. Proximity operators: So near and yet so far. In *Proc. TREC*, pages 131–143, 1995.

[13] B. He, J. X. Huang, and X. Zhou. Modeling term proximity for probabilistic information retrieval models. *J. of Information Sciences*, 181(14):3017–3031, 2011.

[14] S. Huston and W. B. Croft. A comparison of retrieval models using term dependencies. In *Proc. CIKM*, pages 111–120, 2014.

[15] S. Huston, J. S. Culpepper, and W. B. Croft. Indexing word sequences for ranked retrieval. *ACM Trans. Inf. Sys.*, 32(1):3, 2014.

[16] M. Kaszkiel and J. Zobel. Effective ranking with arbitrary passages. *J. Amer. Soc. Inf. Sc. Tech.*, 52(4):344–364, 2001.

[17] L. Lee. IDF revisited: A simple new derivation within the Robertson-Spärck Jones probabilistic model. In *Proc. SIGIR*, pages 751–752, 2007.

[18] X. Lu, A. Moffat, and J. S. Culpepper. On the cost of extracting proximity features for term-dependency models. In *Proc. CIKM*, pages 293–302, 2015.

[19] Y. Lv and C. Zhai. Positional language models for information retrieval. In *Proc. SIGIR*, pages 299–306, 2009.

[20] C. Macdonald, I. Ounis, and N. Tonellotto. Upper-bound approximations for dynamic pruning. *ACM Trans. Inf. Sys.*, 29(4):17, 2011.

[21] D. Metzler. *Beyond Bags of Words: Effectively Modeling Dependence and Features in Information Retrieval*. PhD thesis, University of Massachusetts, Amherst, 2007.

[22] D. Metzler. Automatic feature selection in the Markov random field model for information retrieval. In *Proc. CIKM*, pages 253–262, 2007.

[23] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *Proc. SIGIR*, pages 472–479, 2005.

[24] J. Peng, C. Macdonald, B. He, V. Plachouras, and I. Ounis. Incorporating term dependency in the DFR framework. In *Proc. SIGIR*, pages 843–844, 2007.

[25] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. SIGIR*, pages 275–281, 1998.

[26] Y. Rasolofo and J. Savoy. Term proximity scoring for keyword-based retrieval systems. In *Proc. ECIR*, pages 207–218, 2003.

[27] S. E. Robertson. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends in Inf. Ret.*, 3(4):333–389, 2009.

[28] S. E. Robertson and K. Spärck Jones. Relevance weighting of search terms. *J. Amer. Soc. Inf. Sc.*, 27(3):129–146, 1976.

[29] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proc. TREC*, 1994.

[30] R. Song, M. J. Taylor, J.-R. Wen, H.-W. Hon, and Y. Yu. Viewing term proximity from a different perspective. In *Proc. ECIR*, pages 346–357, 2008.

[31] K. M. Svore, P. H. Kanani, and N. Khan. How good is a span of terms? Exploiting proximity to improve web retrieval. In *Proc. SIGIR*, pages 154–161, 2010.

[32] T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *Proc. SIGIR*, pages 295–302, 2007.

[33] A. Trotman, X. Jia, and M. Crane. Towards an efficient and effective search engine. In *Proc. SIGIR 2012 Wrkshp. Open Source Inf. Retr.*, pages 40–47, 2012.

[34] A. Trotman, A. Puurula, and B. Burgess. Improvements to BM25 and language models examined. In *Proc. Aust. Doc. Comp. Symp.*, pages 58–65, 2014.

[35] C. Zhai. *Statistical Language Models for Information Retrieval*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool, 2008.

[36] J. Zhao, J. Huang, and B. He. CRTER: Using cross terms to enhance probabilistic information retrieval. In *Proc. SIGIR*, pages 155–164, 2011.

[37] J. Zobel and A. Moffat. Inverted files for text search engines. *ACM Comp. Surv.*, 38(2):6.1–6.56, 2006.