Inf Retrieval (2014) 17:380–406
DOI 10.1007/s10791-014-9243-x

Distance matters! Cumulative proximity expansions for ranking documents

Jeroen B. P. Vuurens · Arjen P. de Vries

Received: 4 August 2013 / Accepted: 2 July 2014 / Published online: 12 July 2014

Abstract In the information retrieval process, functions that rank documents according to their estimated relevance to a query typically regard query terms as being independent. However, it is often the joint presence of query terms that is of interest to the user, which is overlooked when matching independent terms. One feature that can be used to express the relatedness of co-occurring terms is their proximity in text. In past research, models that are trained on the proximity information in a collection have performed better than models that are not estimated on data. We analyzed how co-occurring query terms can be used to estimate the relevance of documents based on their distance in text, which is used to extend a unigram ranking function with a proximity model that accumulates the scores of all occurring term combinations. This proximity model is more practical than existing models, since it does not require any co-occurrence statistics, it obviates the need to tune additional parameters, and has a retrieval speed close to competing models. We show that this approach is more robust than existing models, on both Web and newswire corpora, and on average performs equal or better than existing proximity models across collections.

Keywords Term dependency · Term proximity · Query expansion

1 Introduction

Information retrieval (IR) has changed considerably over the years, due to the expansion of the World Wide Web and an increased use of the Web as the primary source of information.

J. B. P. Vuurens (✉)
The Hague University of Applied Sciences, The Hague, The Netherlands
e-mail: j.b.p.vuurens@tudelft.nl

J. B. P. Vuurens · A. P. de Vries
Delft University of Technology, Delft, The Netherlands
e-mail: arjen@acm.org

A. P. de Vries
CWI, Amsterdam, The Netherlands

While there are many ways to measure the performance of an IR system, in general, the aim is to rank documents according to relevance perceived by the user. Retrieving relevant documents is a challenging task, because there usually exists no "perfect query" that unambiguously provides a clean match between the user needs and relevant information. In fact, queries typically match more irrelevant documents than relevant ones.

The problem of ranking documents for a given query is usually simplified by treating query terms as being independent, ignoring any user's interest in the joint co-occurrence of query terms such as forming a noun phrase. One feature that can be used to express the relatedness of co-occurring terms is their proximity in text. Intuitively, researchers have suspected that query terms that appear closer together in documents represent stronger evidence for relevance (Keen 1991; Croft et al. 1991; Clarke et al. 2000; Metzler and Croft 2005; Tao and Zhai 2007; Zhao and Yun 2009; Lv and Zhai 2009). Early studies used theoretical assumptions about proximity, which provided occasional rather than substantial results, e.g. (Fagan 1987; Croft et al. 1991). More recent proximity models make use of parameters that allow them to fit themselves to the data in the collection and more consistently improve retrieval performance over independent term baselines (Metzler and Croft 2005; Tao and Zhai 2007). Although these studies show the potential of term proximity for IR, they provide limited insight into the relation between term proximity and document relevance. To obtain optimal results, these models often use only a selection of term combinations and occurrences within some maximum word distance. As a consequence, there is great diversity between studies upon their assumption how relevance is affected by the distance between query terms (Clarke et al. 2000; Metzler and Croft 2005; Tao and Zhai 2007; Zhao and Yun 2009; Lv and Zhai 2009).

We hypothesize that the performance of proximity retrieval models can be improved by removing constraints regarding distance and the selection of term combinations. In this study, we specifically reexamine the relationship between the distance of co-occurring query terms and their estimated relevance, to design a cumulative proximity expansions (CPEs) retrieval model. We show that this model is more robust and performs equal or better to state-of-the-art proximity models on both Web and newswire collections. This model is also more practical than existing models, since it does not require any co-occurrence statistics, it obviates the need to tune additional parameters, and has a retrieval speed close to competing models.

The paper is structured as follows: Sect. 2 discusses related work that is relevant to term dependency in IR models. In Sect. 3, the effects of promoting proximity are examined. Based on this analysis, we describe a proximity model in Sect. 4. In Sect. 5, we discuss the implementation, which is available for download, and compare the retrieval speed of the proposed model to five baselines over eleven test sets. Section 6 presents the empirical results of the model over ad-hoc TREC collections and the comparison with state-of-the-art proximity models. The conclusions are presented in Sect. 7.

2 Related work

This section discusses previous research in the area of term dependencies, specifically: query operators to match term-dependencies in documents, the selection of term combinations used in a dependency model, the relation between the proximity of query terms and the likelihood of being relevant, and how proximity evidence can be used in a function to rank documents.

We first survey, briefly, the literature addressing term proximity for IR models. We then summarize the key aspects with respect to the question of scoring term proximity. The third and final subsection details the baseline approaches chosen for our own empirical work.

2.1 Term dependency

Term dependency is a recurring topic throughout the history of IR research, and it is plain impossible to do justice to every single contribution in this area. As we will see, many attempts never materialized in consistent improvements in retrieval performance. However, with today's larger corpora, improvements are feasible and there has been a renewed interest in this topic. We highlight the papers most relevant for our own work in this section.

An early paper to address the effect of term dependency on text retrieval is Van Rijsbergen (1977), describing a theoretical basis for the use of co-occurrence data in ranking documents. For every query term, the most likely dependent other query term is selected based on the expected mutual information measure computed for the pair of terms. Van Rijsbergen proposes to use the resulting maximum spanning tree (MST) for retrieval. Only years later, this idea has been followed up upon when Nallapati and Allan (2002) presented their approach to select the most significant query term dependencies. To reduce the runtime required, the MST was build using sentence statistics rather than document statistics. Results of this study were characterized as "a slight improvement in performance", and we are not aware of later follow-up work.

The rather early empirical study of Fagan (1987) discussed a comparison of methods to select phrases with semantically related terms. Fagan's most successful attempt matched pairs of query terms to document contents, but without consistent improvements in retrieval performance.

Keen (1991) describes results of another early empirical study, using test collection created in 1982 from bibliographic summary records. Keen suggested that using information on term positions can help narrow down search results, by screening out irrelevant results. As far as we know, he was the first to formulate the intuition that motivated our own exploration of term proximity, when he suggested the number of *intervening terms* as the factor determining the strength of the relation between pairs of query terms: "the number of non-matching terms found to lie between the first and last matching terms in the sentence". He explored seven different approaches to make use of the proximity between terms, based on the actual term distance, query term co-occurrence in sentences, and a combination of these two principles (e.g. terms in close proximity within a sentence). All seven methods were demonstrated experimentally to improve retrieval performance over the (by now outdated) baseline, a system using coordination level ranking. In this study, the most effective results were obtained with an algorithm that rewards a low distance between query term pairs.

Croft et al. (1991) first explored how to integrate term co-occurrence information derived from phrases into the inference network retrieval model underlying InQuery (and, much later, Indri). The noun phrases considered were extracted from the information request using a stochastic tagger. Croft et al. further suggested the idea of removing individual query terms with a high collection frequency, and matching these only as constituents of a phrase, e.g. the word "system" in "computer system" and "operating system". They obtained improvements in precision at low recall levels, but results at higher levels of recall were inconclusive. The authors noted that their results suggest a higher contribution to retrieval performance on larger collections, a finding that has been confirmed in later work, using a new retrieval model based on Markov random fields (Metzler and Croft 2005).

Various more recent studies have introduced term dependencies in the language modeling approach to IR. Song and Croft (1999) first considered extending the "standard" unigram models by interpolation with a model for bi-grams. A small scale experiment

indicated improved retrieval effectiveness by using the word pairs. Gao et al. (2004) proposed the Dependence Language Model, a joint distribution between a unigram language model and a dependency model that promotes the documents that contain co-occurring query terms (within a sliding window of three words). From a training corpus they estimated the most likely "linkage" that sequentially connects all query terms using every query term once. Only term-pairs in this linkage were considered in the dependency model. Their model consistently improved retrieval performance on smaller TREC collections; however, according to Metzler and Croft (2005) and He et al. (2011), the requirement to compute the likelihood of all possible link structures for a query may be prohibitive for application in practical retrieval systems.

A different line of research explored how to integrate term dependencies in the classic probabilistic retrieval model. For example, Rasolofo and Savoy (2003) proposed to expand BM25 with a proximity measure, by accumulating a distance score for every co-occurring term pair within a sliding window of five words. Their accumulated distance function replaces the term frequency in a BM25-like function, using the lowest weight of the two terms in the pair. The score of the co-occurring term pairs is then added to the score of the unigrams. Their experimental results show improvements at the top of the ranking (precision at five), for all three test sets used, but with mixed results on other metrics. Song et al. (2008) introduced a different perspective on term proximity, by forming non-overlapping spans of multiple query terms (not just query term pairs). Each span (identified through intuitive heuristic rules) is then assigned a relevance weight, based on the length of the span and the number of query terms contained. These weights are aggregated per query term, replacing the original term frequency of the BM25 ranking formula. He et al. (2011) used a sliding window to count the frequency of n-grams in a document, and modified BM25 to score n-grams containing multiple query terms in a way similar to unigrams. In their survival model, they promoted term dependency using the minimal number of words that separate a sequence that contains all query terms.

Metzler and Croft (2005) introduced the Markov random field (MRF) retrieval model, a flexible model that is especially suited for modelling term dependencies. The MRF is constructed from a graph where nodes correspond to query terms, and the edge configuration imposes the independence assumptions made. The authors present three variants, corresponding to the traditional full independence, a new sequential dependence model (SDM) where neighboring query terms are connected by an edge, and a fully connected variant (where every query node is connected to every other query node). Potential functions defined over the cliques in these graphs determine the final ranking function; that combines linearly a relevance score for independent query terms with a score for ordered term pairs (in the sequential dependence case) and one for unordered term combinations (in the full dependence case). The mixture parameters are tuned by cross-validation [selecting the highest mean average precision (MAP) for each fold]. Empirical results show significant improvements by modelling term dependencies explicitly in the MRF. The authors concluded that the SDM is the best choice on smaller but homogeneous collections, with longer queries, while the full dependence model (FDM) attained better results for larger, less homogeneous collections, but using shorter queries.

Metzler and Croft (2007) later expanded the MRF framework with a method to select the most likely "latent concepts" that the user had in mind, but did not express in the query. Similar to the relevance model by Lavrenko and Croft (2001) single word concepts are extracted from (pseudo-) relevance feedback documents, to which they add extracted multi-word concepts. The expansion with latent concepts improves the performance significantly over the original MRF model.

Shi and Nie (2010) reflected on Metzler and Croft (2005), arguing that it is not reasonable to expect the same fixed value to score unigrams, term sequences and term co-occurrences within a sliding window. Both Shi and Nie (2010) and Bendersky et al. (2010) therefore extended Metzler and Croft's SDM model with separate parameters for each unigram and each term-pair. This however leads to a huge number of model parameters that require tuning through n-fold cross-validation, for which both approaches build on a coordination-ascent search algorithm, while Shi and Nie (2010) combines this with an approach based on Support Vector Machine regression. Bendersky and Croft (2012) then proposed a model that is reminiscent to MRF, in which three types of linguistic structures are considered: the original query terms, adjacent term-pairs and unordered co-occurrences of selected term combinations within some window size. The ranking function combines independent scoring of all concepts using a log-linear scoring function, extended with the score of a "global hyper edge" that considers each concepts contribution to the entire set of query concepts. A possible advantage of the global hyper edge over the FDM by Metzler and Croft (2005) is that it can express a dependency between multi-term concepts, rather than all single terms co-occurring within some distance. A learning-to-rank approach was used to train the large set of parameters. Unfortunately, the improvements in effectiveness were only marginal.

Two fairly recent papers aimed to investigate ranking using term proximity, but far less tightly connected to the derivation of the retrieval model in which the term dependencies are to be integrated. First, Büttcher et al. (2006) proposed what they called a *cumulative model* that calculates separate proximity scores per query term. Their algorithm considered especially its implementation in the inverted file indexing structure that forms the core component of virtually all retrieval systems at the time. While traversing the posting lists considered for a given query, whenever a query term is encountered that is different from the query term last seen, a distance score is added to their respective accumulators. The scores per term are eventually computed as separate evidence, by using the proximity score instead of the term frequency. A similar approach was taken in Tao and Zhai (2007). The authors compared five proximity measures, each returning an aggregated outcome per document, which is then converted into a term weight using a convex decay over the distance function, and added to either the KLD or BM25 retrieval scores. The best performing proximity measure was the minimum distance between any two different query terms occurring in the text, which consistently improved retrieval performance and was shown to give results comparable to those obtained in Metzler and Croft (2005).

Building upon this latest work, Zhao and Yun (2009) presented the proximity language model (PLM). Here, the minimum distance metric of Tao and Zhai (2007) is used between all the query term pairs. The sum of minimal distances is converted into a score that is added directly to each unigram's term frequency in the KLD function. Their results show a higher mean average precision when compared to those of Tao and Zhai, however, on more than 50 % of the reported experiments the results did not significantly improve over the KLD baseline.

A few more approaches should still be mentioned, even if this brief discussion can never do justice to the complete history of term dependency in IR.

Lv and Zhai (2009) proposed the positional language model, that builds upon the idea to propagate each occurring term over the word positions in the document (a notion first introduced in (De Kretser and Moffat 1999). The authors use kernel density estimation, to essentially create a separate language model for every word position in a document. Their ranking function therefore initially ranks document positions instead of documents.

Most learning to rank (LTOR) approaches use features derived from query term co-occurrences. As an example, Cummins and O'Riordan (2009) used machine learning to develop term-term proximity functions that optimize mean average precision. They considered aggregated distance statistics, such as the average and minimal distance between query terms in the document. In spite of achieving a consistent increase in MAP on the test collections, the improvements were not always found to be statistically significant. In general, the retrieval functions resulting from the machine learning process tend to exceed human comprehension, not really helping us to understand how distance and relevance are actually related; merely confirming the intuition that such a relationship exists.

So far, we have primarily focused on the way term proximity is integrated in retrieval models for document retrieval. Term proximity is of course closely related to topics like passage retrieval, which in general divides documents into passages as the basis for ranking (Liu and Croft 2002; Tellex et al. 2003). Similar to term proximity, passage retrieval is less likely to promote documents in which the terms appear further apart, but has been criticized by Tao and Zhai (2007) for being more coarse and limited. Other related approaches are XML information retrieval and sentence retrieval, that both capture aspects of term proximity in a similar way to the passage retrieval approaches.

Finally, term proximity can be used in different aspects of a retrieval system, notably by improving the selection of query expansion terms. E.g. Vechtomova and Wang (2006) compared distance functions to improve the selection of query expansion terms. Empirically, the best variant used an inversely proportional function over term distance to rank candidate expansion terms, which outperformed the use of exponential and logarithmic distance functions. In recent work, Miao et al. (2012) also used proximity to improve the selection of expansion terms from feedback documents. They rank candidate terms using word distance with query terms in the feedback documents, using the hyperspace analogue to language (HAL) model. The results consistently improved performance over a baseline of BM25 with Rocchio pseudo-relevance feedback.

2.2 Design aspects of proximity models

A large variety of approaches of dealing with term proximity in IR has been presented throughout the years. Looking back on all these works, however, the following three research questions should still be considered as unanswered:

1. What is the range within which co-occurrences of query terms should be considered?
2. How should the distance between co-occurring terms be reflected in their respective term weights?
3. How should evidence derived from term proximity be integrated in the retrieval model?

Nearby term co-occurrences have generally been considered to provide stronger evidence of relevance than more distant co-occurrences. The majority of studies has only considered occurrences within short distance from each other. Given that the relation between relevance and the co-occurrence of query terms is not obvious, let alone when these occurrences are far apart, this seems a natural choice. When asking at what range query term co-occurrences may still influence the relevance estimation process, the literature does not provide an answer. Studies like (Croft et al. 1991; Rasolofo and Savoy 2003; Metzler and Croft 2005) have only considered terms that co-occur closely, usually within a window of size eight to ten. However, Song et al. (2008) reported an improvement in retrieval effectiveness using a much larger window size of 50. Similarly, Bendersky and

Croft (2012) observed that the distance between the terms (that together constitute a higher level concept) may span a much greater distance than the typical sliding window considered in most of the research.

The second open question concerns how to weigh terms that occur in the document using proximity information. A few studies have simply used a constant contribution, irrespective of the distance between the query terms considered (e.g. Fagan 1987; Croft et al. 1991; Metzler and Croft 2005). Many researchers have however introduced a method to discount the contribution of co-occurrences based on their distance in text. Table 1 summarizes how the distance between query term co-occurrences has been weighted. Here, *span* refers to the number of word positions covered; $\mathscr{K}$ a threshold on span size above which weights are lowered; *windowsize* the maximum span within which co-occurrences are scored; *terms* the number of query terms that make up the co-occurrence; N is the last position in the document; and, i is the word position for which the weight is estimated. The weight accumulates the distance $|i-j|$ between the current position i and query term occurrences j using a Gaussian kernel. For further details, please refer to the original paper Lv and Zhai (2009). The symbols x, y, α, *para* and σ are free parameters that need to be tuned on the document collection used. The functions considered have in common that they are convex and monotonically decreasing. Most studies have assigned a default score of one to adjacently appearing terms, with the exception of Tao and Zhai (2007), Song et al. (2008) and Lv and Zhai (2009). Which of these functions would be the preferred choice has never been answered satisfactory.

The first two questions may be left unanswered, or only answered implicitly, when relying on a machine learning method to estimate the best choice given training data to tune model parameters. Then, instead of positing a general assumption on how proximity relates to relevance, the scoring function can be adjusted to the collection, without the need to making such choices a priori. Examples include Metzler and Croft (2005), Tao and Zhai (2007) and Zhao and Yun (2009); SDM for example distinguishes between contiguous and non-contiguous appearance of terms in a query and document, and estimates the weights to combine these using cross-validation. Tao and Zhai (2007) use a parameter in a convex decay over distance function, allowing the model to adapt the function to more optimal performance.

Table 1 Various functions for determining the weight of a co-occurrence based on the distance between terms

$weight = \begin{cases} 1 & \text{if span} < \mathscr{K} \\ \frac{\mathscr{K}}{span}, & \text{otherwise.} \end{cases}$	Clarke et al. (2000)
$weight = \frac{1}{\sqrt{span-1}}$	Hawking and Thistlewaite (1995)
$weight = \frac{1}{(span-1)^2}$	Rasolofo and Savoy (2003)
$weight = windowsize - span + 1$	Miao et al. 2012)
$weight = \left(\frac{terms}{span}\right)^x \cdot terms^y$	Song et al. (2008)
$weight = \log\left(\alpha + e^{-min(span-2)}\right)$	Tao and Zhai (2007)
$weight = para^{-min(span-2)}$	Zhao and Yun (2009)
$weight_i = \sum_{j=1}^{N} exp\left[\frac{-(i-j)^2}{2\sigma^2}\right]$	Lv and Zhai (2009)

Instead of following this trend to let the model adapt to training data, this paper does actually attempt to give an explicit statement of how term co-occurrences is expected to influence relevance. Assuming the first two questions can be settled, the proximity evidence still needs to be combined with the other sources of information upon which a document's probability of relevance is estimated, and especially the independent query term occurrences.

The two most common design patterns in the literature surveyed have (1) added proximity evidence directly to the independent query term frequencies, using the original retrieval model *as is* (see e.g. Svore et al. 2010; Zhao and Yun 2009), or (2) scored proximity evidence separately from the other relevance information, mixing the proximity based score with that of the independent term occurrences (see e.g. Metzler and Croft 2005; Tao and Zhai 2007). Consider the following example. Let query terms "Albert Einstein" occur once and adjacently in two different documents A and B, but with different frequencies for the word "Einstein". If proximity evidence is added to the raw unigram counts, the document with the lowest unigram count will gain the most from adding the co-occurrence information. When viewed as two separate sources of relevance information, both documents would receive the same contribution for the evidence of query term co-occurrences, irrespective of the frequency of the individual query terms.

2.3 Proximity baselines

In this study, we will compare the retrieval performance of four proximity baselines that can be considered state-of-the-art based on the results presented. The selected baselines score terms that occur in a document independently using Dirichlet-smoothed language model functions that are rank equivalent to each other. Therefore, comparing the results of each model to the independent term baseline, will reveal how effective each model is in additionally using proximity information.

Zhai and Lafferty (2004) propose to use negative Kullback–Leibler divergence between a query language model and a Dirichlet smoothed language model of a document (KLD), which they reformulated by removing the query entropy which does not affect document ranking. In Eq. (1), documents D are ranked for a query Q, q_i is a term in $Q, tf_{q_i,D}$ is the frequency of q_i in $D, |D|$ is the number of terms in D, and μ is the Dirichlet smoothing parameter. In Eq. (2), cf_{q_i} is the frequency of q_i in the collection C and $|C|$ is the number of words in the collection.

$$KLD(Q,D) \equiv \sum_{q_i \in Q} \left[\log\left(1 + \frac{tf_{q_i,D}}{\mu \cdot P(q_i|C)}\right) + \log\frac{\mu}{\mu + |D|} \right] \tag{1}$$

$$P(q_i|C) = \frac{cf_{q_i}}{|C|} \tag{2}$$

Tao and Zhai (2007) presented a simple baseline for scoring term dependency. Using Eq. (3), documents are ranked according to the sum of KLD over independent query terms and a proximity function $\pi(Q,D)$. In Eq. (4), α is a free parameter and $\delta(Q,D)$ is a distance function, for which Tao and Zhai experimented with five variants. In Eq. (5), $Dis(q_i,q_j;D)$ is a function that returns the minimum distance in word positions between all occurrences of query terms q_i and q_j in D, or, $|D|$ if D does not contain both terms. In other words, δ is defined as the minimum distance between any two occurrences of distinct query terms, the variant which provided the best results in the retrieval experiments of Tao and Zhai.

$$MinDist(Q,D) = KLD(Q,D) + \pi(Q,D) \tag{3}$$

$$\pi(Q,D) = \log\left(\alpha + e^{-\delta(Q,D)}\right) \tag{4}$$

$$\delta(Q,D) = \min_{q_i,q_j \in Q \cap D, q_i \neq q_j} \left\{Dis(q_i,q_j;D)\right\} \tag{5}$$

Zhao and Yun (2009) presented the PLM (Eq. 6), in which they re-estimate the seen word probability $P(q_i|D)$ with respect to the proximity model (Eq. 7), and assign $\alpha_D \cdot P(q_i|C)$ to unseen terms (Eq. 6). In their model, the proximity factor adjusts the seen word probability in a document by adding a proximity centrality $Prox_B$ to the term count in the document (Eqs. 7, 8). Besides the symbols already defined for KLD and MinDist, in Eq. (9), $Dis(q_i,q_j;D)$ returns the minimal distance between all occurrences of term q_i and q_j in D as described by Tao and Zhai (2007), and λ and *para* are free parameters.

$$PLM(Q,D) = \sum_{q_i \in Q \cap D} P(q_i|Q) \cdot \log \frac{P(q_i|D)}{\alpha_D \cdot P(q_i|C)} + \log \alpha_D \tag{6}$$

$$P(q_i|D) = \frac{tf_{q_i,D} + \lambda \cdot Prox_B(q_i;D) + \mu \cdot P(q_i|C)}{\mu + |D| + \lambda \cdot \sum_{q_i \in Q} Prox_B(q_i;D)} \tag{7}$$

$$\alpha_D = \frac{\mu}{\mu + |D| + \lambda \cdot \sum_{q_i \in Q} Prox_B(q_i;D)} \tag{8}$$

$$Prox_B(q_i;D) = \sum_{q_j \in Q, q_i \neq q_j} \text{para}^{-Dis(q_i,q_j;D)} \tag{9}$$

Metzler and Croft (2005) proposed to estimate the relevance of documents using the Markov random fields framework. A Markov random field is constructed from a graph in which nodes (random variables) correspond to the document and the query terms, and edges represent dependencies between these random variables. Due to conditional independence, the joint distribution between these variables can be factored by considering only the cliques in this graph. Therefore, scoring a document boils down to considering separately the sets T, O, and U, of, respectively, query terms treated as independent (T), contiguous query terms (O), and otherwise dependent query terms (U).

Metzler and Croft consider two term dependency variants. In the SDM, only adjacent query terms are considered directly dependent, and O and U will consist of all pairs of terms that appear adjacently in the query. FDM, all query terms are directly dependent upon each other, O consisting of all contiguous sequences of two or more query terms and U of all combinations of two or more query terms. The resulting ranking function in Eq. (10) combines the scores for these three sets of cliques using the linear mixture parameters λ_T, λ_O and λ_U.

The independent query terms T are scored using function f_T (Eq. 11), which is a language modeling estimate smoothed by a Dirichlet prior α_D, where $tf_{q_i,D}$ is the frequency of term q_i in document D, $|D|$ is the number of terms in D, cf_{q_i} is the frequency of the term in the collection, and $|C|$ the total number of terms in the collection. Contiguously appearing query terms O are scored using f_O (Eq. 12), in which $tf_{\#1(q_i,\ldots,q_j),D}$ denotes the number of times the sequence of terms $q_i,\ldots,q_j$ appears contiguously in D, and $cf_{\#1(q_i,\ldots,q_j)}$ in the collection. Combinations of query terms U are scored using f_U (Eq. 13), in which $tf_{\#uwN(q_i,|\ldots,q_j),D}$ is the number of times all query terms in the clique

appear in any order within a window of N words in D, with N set to four times the number of terms in the clique, and $cf_{\#uwN(q_i,\ldots,q_j)}$ the frequency of that event in the collection. In Eq. (14), α_D is the smoothing parameter also described by Zhai and Lafferty (2004). By definition, $\lambda_T = 1 - \lambda_O - \lambda_U$ leaving λ_O, λ_U and μ to be trained as free parameters.

$$\begin{aligned} MRF(Q,D) = \lambda_T \sum_{(q_i,D)\in T} f_T(q_i,D) \\ + \lambda_O \sum_{(q_i,\ldots,q_j,D)\in O} f_O(q_i,\ldots,q_j,D) \\ + \lambda_U \sum_{(q_i,\ldots,q_j,D)\in U} f_U(q_i,\ldots,q_j,D) \end{aligned} \tag{10}$$

$$f_T(q_i,D) = \log\left((1-\alpha_D)\frac{tf_{q_i,D}}{|D|} + \alpha_D \frac{cf_{q_i}}{|C|}\right) \tag{11}$$

$$f_O(q_i,\ldots,q_j,D) = \log\left((1-\alpha_D)\frac{tf_{\#1(q_i,\ldots,q_j),D}}{|D|} + \alpha_D \frac{cf_{\#1(q_i,\ldots,q_j)}}{|C|}\right) \tag{12}$$

$$f_U(q_i,\ldots,q_j,D) = \log\left((1-\alpha_D)\frac{tf_{\#uwN(q_i,\ldots,q_j),D}}{|D|} + \alpha_D \frac{cf_{\#uwN(q_i,\ldots,q_j)}}{|C|}\right) \tag{13}$$

$$\alpha_D = \frac{\mu}{\mu + |D|} \tag{14}$$

A final remark on the relationship between the scoring of independent terms in the MRF and KLD baselines. Although the function used to score the independent query term occurrences has a different form in Eq. (10) than the KLD function described in Eq. (1), both functions are Dirichlet smoothed language model estimates and are in fact rank equivalent if used with the same value for μ (refer to "Appendix 1" for a derivation of this equivalence).

3 Analysis

To examine term proximity, we used an oracle system on the TREC-6 ad-hoc topics to improve the retrieval performance by expanding the queries with proximity operators. We describe how the oracle system maximizes results, report the reformulation for the most improved queries and discuss the results. We then continue to analyze how the relevance of documents can be estimated based on the distance between co-occurring query terms.

3.1 Oracle experiment

Most existing proximity models make limited use of the available proximity information in documents, considering only a selection of term combination and occurrences within some maximum word distance. As far as we know, previous studies do not explain why more distance between co-occurring query terms represent weaker evidence for a document being relevant, or whether it is justified to assume that proximity is only useful within a limited word distance. We hypothesize that words in between query term occurrences weaken their relatedness. Therefore, proximate terms are more likely to

Query: best basketball player	
Document 1	: ... best basketball player ...
Document 2	: ... best training exercise for basketball players ...
Document 3	: ... best basketball for beginning players ...

Fig. 1 The query is more likely to be satisfied by Document 1 than by Document 2 or 3

appear in relevant documents than distant ones. In Fig. 1, we illustrate this intuition by sentences that would each match the hypothetical query "best basketball player". Although in relevant documents these terms do not necessarily appear consecutively, more distance between the query terms provides more opportunity to divert or weaken the relation between them, e.g. "best training exercise for basketball players" or "best basketball for beginning players". Since we expect an increase in the number of intermediate words to increase the likelihood of weakening the relation between query terms, the relation between proximity and relevance may neither be limited to some distance, nor only apply to some term combinations.

To analyze the potential of using proximity in ranking documents, we constructed an oracle system which performed a simple breadth-first search to optimize queries by adding proximity operators using two or more query terms, evaluating each variant using known relevance judgments. The system used all possible query-term combinations as potential proximity expansions, and was allowed to independently adjust each proximity operator by setting a *weight* $\in \{1.0, 0.50, 0.25\}$ and a maximum *span* $\in \{2, 3, 4, 5, 10, 20, 50, 100, 200, 500\}$. The original query was used as the initial best query. For this query, all previously unseen single variations were tried, i.e. adding, removing or modifying only one proximity expansion. The best query was then replaced by the variant with the highest mean average precision (or set of variants when tied) and used as input for the next iteration to try new variants. This was repeated until there were no more untried single variations to the best query, thus converging to a (local) optimum.

We used the oracle system to improve the TREC-6 queries with proximity expansions, and list the 10 queries that were improved most in Table 2. Documents were scored according to Eq. (15), which uses KLD to score the independent terms, and KLD_{co} to score the proximity expansions M. In Eq. (16), m_i is the ith proximity expansion in M, λ_i is the weight for m_i, $tf_{\#uw\delta_i(m_i),D}$ is number of occurrences matched by m_i in document D, matching the terms in any order within a window of δ_i word positions. In Eq. (17), $cf_{\#uw\delta_i(m)}$ counts the occurrences of m_i in the entire collection and $|C|$ is the number of word in the collection. To explain syntax used in Table 2, "{air pollution span = 200}#0.5" scores all unordered co-occurrences of "air" and "pollution", using $\delta = 200$ and $\lambda = 0.5$. Over the TREC-6 test set, the oracle system improved over the KLD baseline by 17.8 % and over the SDM baseline by 13.3 %, indicating the potential contribution for term proximity to estimate document relevance exceeds that of the SDM baseline. We make three observations: (1) for queries with three or more terms, the best solution often uses several overlapping proximity operators, e.g. "undersea fiber cable" along with "fiber cable"; (2) using a maximum allowed span of more than 100 words is most effective for some term combinations; (3) the best expansion often uses lower weights for the co-occurrences than for the original query terms.

Table 2 10 TREC-6 queries that gained most (in AP%) in the Oracle experiment

Topic	Original query	Oracle expansion	KLD AP	SDM AP	Oracle AP
303	Hubble telescope achievements	{Hubble telescope span = 2} {Hubble achievements span = 5}	0.205	0.231	0.342
308	Implant dentistry	{Implant dentistry span = 100}#0.25	0.480	0.479	0.554
310	Radio waves and brain cancer	{Radio waves cancer span = 100}#0.25 {Brain cancer span=2}#0.5	0.160	0.185	0.242
311	Industrial espionage	{Industrial espionage span = 10}#0.5	0.372	0.576	0.635
320	Undersea fiber optic cable	{Undersea fiber span = 2}#0.5 {Undersea optic span = 10} {Fiber cable span = 2}#0.5 {Undersea fiber cable span = 3}#0.25 {Optic cable span = 20} {Undersea optic cable span = 5} {Fiber optic cable span = 3}#0.25	0.023	0.028	0.138
329	Mexican air pollution	{Mexican air span = 500} {Mexican pollution span = 500} {Air pollution span = 200}#0.5 {Mexican air pollution span = 20}	0.141	0.125	0.304
331	World Bank criticism	{World Bank span = 2} {World criticism span = 200}#0.25 {World Bank criticism span = 20}#0.5	0.213	0.287	0.419
332	Income tax evasion	{Income tax span = 2}#0.25 {Income evasion span = 10} {Tax evasion span = 2} {Income tax evasion span = 3}	0.126	0.139	0.342
341	Airport security	{Airport security span = 50}	0.232	0.282	0.329
350	Health and computer terminals	{Health computer span = 200}#0.5 {Computer terminals span = 2} {Health computer terminals span = 500}	0.105	0.116	0.169

$$Score(Q, M, D) = KLD(Q, D) + KLD_{co}(M, D) \tag{15}$$

$$KLD_{co}(M, D) = \sum_{i=1}^{|M|} \lambda_i \cdot \log\left(1 + \frac{tf_{\#uw\delta_i(m_i),D}}{\mu \cdot P(m_i|C)}\right) \tag{16}$$

$$P(m_i|C) = \frac{cf_{\#uw\delta_i(m_i)}}{|C|} \tag{17}$$

3.2 The relation between distance and relevance

In Sect. 2.2, we reviewed existing proximity functions and noticed that in general a value of one is assigned to adjacently appearing terms, occurrences that appear non-contiguously in text receive a lower value based on a linear-convex function over their span in text, and

co-occurrences whose span exceeds a maximum distance are ignored. To design a proximity function that benefits from using co-occurrences over longer distance, we argue that the score of both distant and nearby occurrences should reflect their probability to occur in a relevant document. Beeferman et al. (1997) have shown that the co-occurrence frequency between words decays exponentially over their distance. In this study, we are not interested in how often terms co-occur, but rather in the likelihood that the document they occur in is relevant. We analyzed this in a similar experiment, by using all possible combinations of two or more terms from the TREC 1–3 and 5–8 ad-hoc queries, which were counted separately for relevant and irrelevant documents in the corresponding corpora using the TREC qrels. We estimate the likelihood that an occurrence appears in a relevant document given the number of separating terms using Eq. (18), for which we define $d = R$ as a predicate to test that document D is relevant for the given query, m is a combination of two or more terms from query Q as defined by the Power Set $\mathscr{P}_{>1}(Q)$, and the function $tf_{\#ew\delta(m,d)}$ counts all non-overlapping unordered occurrences of m that are separated by exactly δ terms in D. The obtained frequencies were smoothed using a Gaussian kernel with a bandwidth of $1 + (\delta/2)$.

$$P(d = R|\delta) = \frac{\sum_Q \sum_{d \in C: d=R} \sum_{m \in \mathscr{P}_{>1}(Q)} tf_{\#ew\delta(m,d)}}{\sum_Q \sum_{d \in C} \sum_{c \in \mathscr{P}_{>1}(Q)} tf_{\#ew\delta(m,d)}} \tag{18}$$

In Fig. 2, the "all combinations" line is the result of this experiment, which unexpectedly increases monotonically after reaching a minimum value. Inspection of the data revealed that some term combinations exhibit the opposite behavior, such that close proximity is inversely proportional to the probability of relevance. Ideally, we would like to be able to a-priori identify term combinations that have this inverse distance/relevance relationship, but this is not an issue we will resolve in this study. However, we expect that the undesirable effect of these inverse combinations is partly suppressed by the presence of other query terms, which provide a context within which the inverse combinations may behave differently. In order to focus on term combinations that are more relevant at closer distance, we simply identified inversely related term combinations as having a higher average probability of relevance at 500–1,000 terms distance than for 200–500 terms distance. For the set that is labeled "proportional" in Fig. 2, these inversely related term combinations were removed. For the "proportional" set, the probability of appearing in a relevant document over the distance between terms can be approximated with an inversely proportional function, which is drawn as a dotted line in Fig. 2 with a goodness-of-fit of $R^2 > 0.99$. From these results, we make two observations: (1) adjacently appearing query terms are roughly twice as likely to appear in relevant documents than query terms that are far apart, and (2) the probability of appearing together in a relevant document can be estimated using the distance between terms using a function of the order $1/distance$.

4 Methods

In Sect. 2.2, we reviewed existing proximity functions and concluded that several aspects have remained unresolved. In Sect. 3.1, we analyzed the potential of proximity operators to improve retrieval performance and estimated the likelihood that a document is relevant given the word distance between occurring term co-occurrences. We now continue to design a proximity retrieval model that uses these aspects. A special variant of this proximity retrieval model also considers the stop words, which are often ignored by retrieval models.

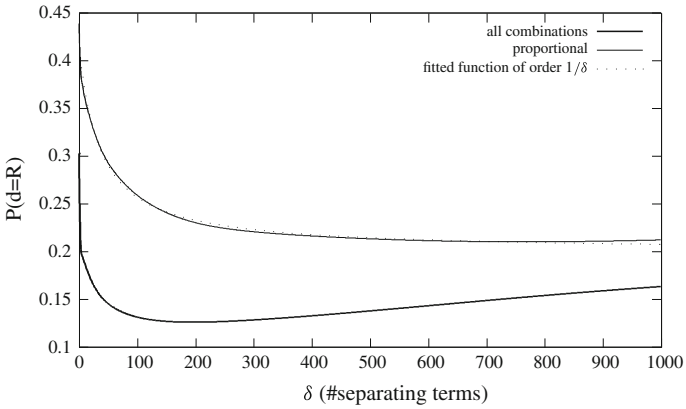

Fig. 2 The probability of term combinations to appear in a relevant document given the number of separating words, estimated from the TREC 1–3, 5–8 ad-hoc topics. The results are compared between "all combinations" and "proportional" term combinations that are more likely to appear in a relevant document at close proximity. Fitted to the proportional set is a function of the order $1/\delta$

4.1 Cumulative proximity expansions model

The cumulative proximity expansions model (CPE) we propose is a simple and, as we will show, effective retrieval model. CPE expands the KLD function for independent query terms with a proximity model that scores every possible combination of two or more query terms. To identify occurrences of term combinations in a document, a proximity operator is used that returns the maximum number of the shortest possible non-overlapping text passages that contain all specified terms in any order. For overlapping occurrence candidates, the shortest or left most candidate is selected first. Different term combinations are scored independently of each other, therefore occurrences of different term combinations can have overlapping word positions, e.g. occurrences of the combination "dog law enforcement" can overlap with occurrences of the combination "law enforcement", which results in a higher estimated relevance of a document in which subsets are co-occurring more closely.

Based on the dependency we found between the distance of co-occurring query terms and the likelihood to appear in a relevant document in Sect. 3.2, we formulate the following requirements to a function to score term co-occurrences: (1) double the score contribution of query terms occurrences if they appear contiguously in a document, similar to the observed likelihood in Fig. 2, (2) let the score increment of non-contiguous query term combinations in documents decay with a function of the order 1/distance, and (3) assign the same score for occurrences that are relatively separated to the same extent. The third requirement helps to normalize the scores between term-combinations of different sizes; to clarify, the relative separation of a 2-term combination separated by 2 other words should be weighed similar to a 3-term combination separated by 4 other words in text, having the same 'density'. In Eq. 19, we score a term combination m for document D, by using a function similar to the KLD function over the contained terms. We estimate the frequency $tf_{m,D}$ in Eq. 20 using a proximity operator $\#uw(m, D)$ to match all occurrences o of m in D, and aggregate the frequency per term combination using a $1/distance$ function based on the number of terms $|m|$ and the span of occurrence o in the document. This score function therefore satisfies all three requirements.

$$PROX(m,D) = \sum_{q_i \in m} \log\left(1 + \frac{tf_{m,D}}{\mu \cdot P(q_i|C)}\right) \tag{19}$$

$$tf_{m,D} = \sum_{o \in \#uw(m,D)} \frac{|m|-1}{|o|-1} \tag{20}$$

Documents are then ranked according to Eq. (21), which combines the KLD baseline with all combinations of two or more query terms m as defined by the power set $\mathscr{P}_{>1}(Q)$ over query Q from which stop words are removed. To balance the score of the term combinations with respect to the score of independent terms, we introduce a weight function Z, which is necessary to compensate because an additional query term will linearly increase the number of scored independent terms, while the number of scored term combinations grows exponentially. However, we expect the scoring mass of the term combinations to grow as a function over $|Q|$ rather than to grow exponentially, because combinations are less likely to occur as they contain more terms. Empirically, we found that Eq. (22) gives stable and close to optimal results.

$$CPE(Q,D) = KLD(Q,D) + \frac{1}{Z} \sum_{m \in \mathscr{P}_{>1}(Q)} PROX(m,D) \tag{21}$$

$$Z = |Q| \tag{22}$$

4.2 Stop words in the proximity model

The default proximity model (Eq. 21) uses the power set $\mathscr{P}_{>1}(Q)$, where Q is the query from which stop words are removed. However, in proximity models stop words could be more useful than in independent term models, promoting multi-word concepts (e.g. "The Beatles") or passages containing an intended relation nearby targeted terms (e.g. "boy likes girl", "parents against education"). We hypothesize that considering the stop words in the proximity model may improve retrieval performance. We will test this using a variant of CPE called CPES, in which we replace the power set in Eq. (21) with a power set over a query with stop words. The set of term combinations is cleaned by removing combinations with stop words that are less likely to be relevant. For this, we introduce a simple heuristic: a combination of query terms is 'valid' when the stop words considered are used in combination with all terms that connect them in the query to a non stop word on the left and right, or the query boundary if there is no more non-stop-word. For example, "The Beatles on a zebra crossing" would generate besides "Beatles zebra", "zebra crossing", "Beatles crossing" and "Beatles zebra crossing", the following combinations containing stop words: "The Beatles", 'The Beatles zebra", "The Beatles crossing", "The Beatles zebra crossing", "Beatles on a zebra", "Beatles on a zebra crossing", "The Beatles on a zebra" and "The Beatles on a zebra crossing".

5 Implementation

To improve reproducibility, we carried out all experimental evaluation using a general purpose retrieval framework, and make all code publicly available, as described in this section. We then continue to discuss optimizations in the implementation of CPE and CPES, to achieve acceptable speed while scoring all term combinations.

5.1 Open source

To facilitate reuse of retrieval components and to improve reproducibility, we have created a general purpose framework called repository for IR experiments (RepIR), which uses Hadoop to extract features from a collection and store these in a central repository, which can then be easily accessed for analysis and retrieval tasks. To compare CPE against existing models using the same index, all models listed in Table 3 have been implemented in RepIR according to their specification. To reproduce our results, RepIR can be downloaded as open source or as a Maven project from http://repir.github.io/. This repository also hosts the specific implementations for this paper along with configuration files and tuned parameter settings.

5.2 Retrieval speed

The CPE model uses all possible query term combinations to compute a relevance score for documents given a query. Expanding the query with an independent proximity operator for every term combination, will generate an exponential number of elements to score and thus is not feasible for long queries. However, in general, terms co-occur far less frequently than that they occur as independent terms, becoming more rare when more terms are combined. Given these distributional characteristics, retrieving can be inefficient when each document is independently inspected and scored for every term combination, even for combinations that are not present. Since the KLD function assigns a score of zero to terms that are not present, this simplifies handling term combination that are not present by simply not scoring these. By simultaneously traversing all term-position lists for a document in one pass, counting and scoring only co-occurrences that are encountered in the document, the number of unnecessary operations is reduced, especially for long queries. Instead of using separate proximity operators, we therefore implemented CPE as one module, which combines the scoring of all term combinations, to allow document processing as described.

CPES will be considerably less efficient than CPE, not only because it additionally uses stop words but also because stop words have long postings lists. However, we prune the processing of stop words using the heuristic we presented in Sect. 4.2, by monitoring if the document contains all terms needed to score a combination with a stop word and otherwise stop traversing the positions list of that stop word for that document. For example, the stop word "on" in term combination "the Beatles on a Zebra Crossing" needs only be used for documents that also contain the words "Beatles", "a" and "Zebra". Therefore, if we pass

Table 3 Retrieval models implemented in RepIR for this study

Model	Java classes in package io.github.repir.Strategy
KLD	RetrievalModel, ScoreFunctionKLD
CPE	CPERetrievalModel, CPEFeature, ScoreFunctionKLD
CPES	CPESRetrievalModel, CPESFeature, ScoreFunctionKLD
MinDist	MinDistRetrievalModel, MinDistFeature, MinDistScoreFunction
PLM	PLMRetrievalModel, PLMFeature, PLMScoreFunction
SDM	SDMRetrievalModel, SDMTerm, SDMOrderedPhrases, SDMUnorderedPhrases, ScoreFunctionDirichletLM
FDM	FDMRetrievalModel, FDMTerm, FDMOrderedPhrases, FDMUnorderedPhrases, ScoreFunctionDirichletLM

the last occurrence of the word "Beatles" in a document, we can stop traversing the stop words "the", "a" and "on". As a consequence, a document is not even scored if it only contains the stop words in the query but no other terms, for not meeting the criteria for any scorable unit.

We analyzed the feasibility of using all term combinations on long queries, using the TREC-4 ad-hoc descriptions (except topic 225, for which time measurement was heavily affected by excessive garbage collection). The descriptions in TREC-4 on average contain 17 terms (stop words included) and probably contain more stop words and non-discriminative words than real user queries. We must note that our implementation is not well suited for accurate comparison of retrieval speed; not being optimized for interactive retrieval, operating on a public Hadoop cluster which makes time measurements inaccurate, and using a single index with positional postings lists for all models, which is slower than necessary for KLD which does not need term positions. To obtain a reasonable results given these circumstances, retrieval is executed within a single mapper, measuring the time within the mapper between the start and end of retrieval to eliminate as much time lost to Hadoop overhead as possible, and by using the fastest retrieval time per query per model of 100 repeated executions, which is the time that is least likely to be affected by external delays. The collection statistics for term co-occurrences that are required for SDM were pre-collected and read into memory, therefore taking no extra time in this test.

In Fig. 3, we compare the relative retrieval speeds of SDM, CPE and CPES, measured in times slower than KLD. We do not compare against FDM, which we implemented by simple expansion of the query with proximity operators, and thus not feasible on these long queries. We sorted the topics on the relative retrieval speed of CPES. On average, SDM is 1.7 times slower than KLD, CPE is 1.9 times slower and CPES is 4.7 times slower. For example, one of the longest topics in TREC-4 is 211: "How effective are the driving while intoxicated (DWI) regulations? Has the number of deaths caused by DWI been significantly lowered? Why aren't penalties as harsh for DWI drivers as for the sober driver", which contains 16 non stop words and 17 stop words. The fastest retrieval times measured for this topic using KLD was 1.2 s, SDM 3.0 s, CPE 3.0 s and CPES 7.1 s.

6 Results and discussion

In this section, we first describe the test sets, collections and parameter tuning, then compare the empirical results obtained with all models, discuss the robustness of the retrieval models, analyze the effects of limiting the span of used co-occurrences, and lastly present a qualitative analysis.

6.1 Evaluation setup

For this study, the default repository builder in RepIR was used to create separate positional unigram indexes of the document collections as described by TREC for the TREC 1–3, 5, 6, 7–8 ad-hoc tasks, the English section of ClueWeb09 and ClueWeb12 for the Web Track ad-hoc tasks. During extraction, Unicode and HTML special codes were converted to their ASCII equivalents, and all other Unicode and HTML code was removed with the exception of the contents of "alt" attributes and the meta tags "keywords" and "description". The remaining content was lowercased, tokenized and stemmed with an English Porter-2 stemmer. The vocabulary size was reduced for ClueWeb09 and ClueWeb12 by leaving out numbers as well as words with more than 25 characters, and by

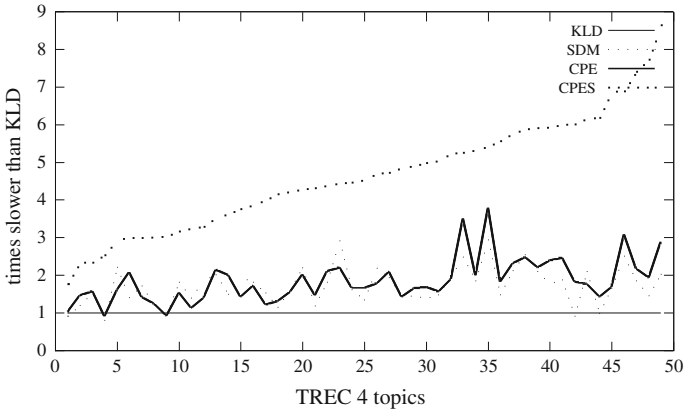

Fig. 3 The relative retrieval speed of SDM, CPE and CPES, measured in times slower than KLD over the TREC-4 ad-hoc descriptions

leaving out infrequent words; i.e. that appear <2 times on TREC 1–8, and <10 times on ClueWeb09 and ClueWeb12. Stop words were indexed thus affecting the positions of non stop word in documents, however, words that are in the SMART list of stop words were not scored during retrieval, except for the experiment that used stop words in the proximity model.

For retrieval, only the titles of the topics were used from the TREC 1, 2, 3, 5, 6, 7, 8 ad-hoc tasks (TREC sets), and the TREC Web Track 2009, 2011, 2012 and 2013 ad-hoc tasks (Web Track sets). Two sets were not used: TREC 4 which contains unrealistic long descriptions as topics for ad-hoc tasks, and Web Track 10, in which 28 out of 50 queries were not useful for evaluation.[1] For the evaluation, all queries in the test sets were used, including the queries that only contain one term, with the exception of queries with no relevant documents in the TREC qrels. Retrieval performance was measured in mean average precision (MAP) for the TREC sets and Statistical Mean Average Precision (StatMAP) for the Web Track sets (Carterette et al. 2008), using the top-1000 retrieved documents.

The free parameters of the proximity models we compare against were tuned using grid search with the following cross-evaluating strategies: leave-a-test-set-out when several test sets share the same collection (TREC 1–3, 7–8, Web Track 9, 11, 12), and tenfold cross evaluation if there is only one test set for a collection (TREC 5, 6, and Web Track 13). The following parameter ranges were scanned to find close to optimal settings: $\mu = 100, 200, \ldots, 5{,}000$, MinDist $\alpha = 0.1, 0.2, \ldots, 1.5$, PLM $\lambda = 1, 2, \ldots, 10$ and $para = 1.1, 1.2, \ldots, 2.5$, and SDM $\lambda_O = 0.04, 0.06, \ldots, 0.20$ and $\lambda_U = 0.02, 0.03, \ldots, 0.80$. Although Metzler and Croft (2005) suggested that the fuzzy matches of unordered window deal better with the noise inherent in Web documents, we did not expect that λ_U would tune to higher settings than λ_T, but on ClueWeb09 $\lambda_U = 0.79$ provided the highest and most stable results. Since the presence of stop words in our index increases the distance between other words, this may have positively affected the importance of matched unordered term combinations. However, it is unclear if this indeed the cause for this anomaly.

[1] These queries contained only one non stop word or did not have any relevant documents in the relevance judgments.

For KLD, we compared the performance when using $\mu = 2{,}000$ as advocated by Zhai and Lafferty (2004) to training using cross-evaluation. On average, the advocated setting performed better, specifically Web Track 2012 was an extreme outlier obtaining +50 % StatMAP for the advocated settings over training using Web Track 2009 and Web Track 2011 combined. Therefore, to obtain a more realistic impression of the benefits of using proximity models on Web collections, we decided to use $\mu = 2{,}000$ for KLD CPE and CPES.

For the ClueWeb collections, the runtime for tuning was significantly reduced by using a smaller subset of the collections, consisting of all documents assessed by TREC and all documents retrieved in the top-10k for all topics and all retrieval models using default parameters. The indexes created for these subsets used the unigram and co-occurrence statistics of the entire collection, so that retrieving the top-1k documents on the subset is near identical to retrieving the top-1k documents on the whole ClueWeb collection. The ClueWeb09 subset contained 1 % of the documents in the entire collection and the ClueWeb12 collection 0.1 %. Checking the results of the parameter tuning, more than 99.9 % of the documents retrieved using the tuned parameters for the Web Tracks sets were contained in the collection used for tuning. Using the tuned parameters, the average difference in StatMAP between retrieving a query's results from the subset using the tuned parameters and from the entire collection corresponded to 0.003 %, caused by a very small number of documents that were retrieved from the full collection after training, but missing from the subset.

6.2 Results

Table 4 presents a side-by-side comparison of CPE and CPES to KLD, MinDist, PLM, SDM and FDM over the TREC sets. On average, FDM and CPE perform better than the other models. The CPES model scores specifically worse than the other models on TREC 1 and 3. We suspect that the effectiveness of using stop words depends on the function these stop words have in the query. In some queries, the stop words are vital clues to recognizing the intended meaning, and thus as a word that should appear in relevant documents. In early TREC editions, topics appear to be phrased as natural language, in which stop words often play a more syntactical role than that they are good predictors of relevant documents. For example, in TREC 3 topic 198: "Gene therapy and its benefits to humankind", the words "and" and "its" may not be here because the user predicts that documents that are relevant contain these words in greater quantity than documents that are not. Using stop words is more likely to help for WT09 topic 42: "the music man", more strongly promoting the intended meaning, which is a musical and not "Music Man" which is a manufacturer of musical instruments.

Table 5 shows a comparison of all models over the Web Track sets. The results show that SDM, FDM, CPE and CPES are relatively more effective than KLD on the Web Track sets than on the TREC sets, which was also suggested by Metzler and Croft (2005). We suspect that the independent term model is negatively affected by the noise and spam that is present in Web collections resulting in a lower baseline, and that the proximity model is less affected by noise and spam by using the distance between co-occurring terms. The results show that FDM performs best on WT09 and WT11, and that CPES outperforms the other models on WT12 and WT13, mostly significant. The behavior of PLM is less predictive on the Web Track sets than on the TREC sets. In Sect. 2.2, we mentioned that PLM adds proximity information to the frequency of seen terms, rather than scoring this as separate evidence, which may explain why this model does not transfer well to Web collections, where document length varies, affecting how proximity is scored. WT09 has

Table 4 Comparison of CPE and CPES to KLD, MinDist, PLM, SDM and FDM over ad-hoc TREC sets

		KLD	MinDist	PLM	SDM	FDM	CPE	CPES
TREC 1	MAP	0.2247	0.2396	0.2432	**0.2436**	0.2433	0.2425	0.2373
	δ (%)		+6.6	+8.2	+8.4	+8.3	+8.0	+5.6
	p value		0.014	0.011	0.026	0.035	0.030	0.103
TREC 2	MAP	0.2065	**0.2145**	0.2128	0.2125	0.2109	0.2142	0.2141
	δ (%)		+3.9	+3.1	+2.9	+2.1	+3.7	+3.7
	p value		0.032	0.071	0.104	0.194	0.026	0.035
TREC 3	MAP	0.2758	0.2872	0.2797	0.2924	**0.2925**	0.2840	0.2786
	δ (%)		+4.1	+1.4	+6.0	+6.0	+3.0	+1.0
	p value		0.007	0.266	0.006	0.004	0.023	0.286
TREC 5	MAP	0.1526	0.1626	0.1790	0.1775	**0.1869**	0.1815	0.1801
	δ (%)		+6.5	+17.3	+16.3	+22.5	+19.0	+18.0
	p value		0.107	0.045	0.076	0.045	0.032	0.039
TREC 6	MAP	0.2280	0.2396	**0.2437**	0.2384	0.2400	0.2411	0.2418
	δ (%)		+5.1	+6.9	+4.6	+5.3	+5.7	+6.0
	p value		0.035	0.007	0.047	0.034	0.020	0.017
TREC 7	MAP	0.1937	0.2073	0.2073	0.2060	0.2071	0.2087	**0.2122**
	δ (%)		+7.0	+7.0	+6.4	+6.9	+7.8	+9.5
	p value		0.024	0.019	0.053	0.042	0.014	0.005
TREC 8	MAP	0.2522	0.2591	0.2567	0.2567	0.2572	0.2621	**0.2633**
	δ (%)		+2.8	+1.8	+1.8	+2.0	+3.9	+4.4
	p value		0.078	0.139	0.205	0.184	0.023	0.014
Average	δ (%)		+5.0	+5.8	+6.1	+6.8	+6.6	+6.1

The percentages represent improvement in MAP over the KLD baseline. These improvements were tested for significance with a paired Student's t-test, one tailed, $\alpha = 0.05$. The highest MAP per test set is in bold

one query in particular on which all proximity models hurt performance: topic 46 "Alexian Brothers Hospital", for which the KLD baseline retrieves three relevant near-identical documents ranked 2–4, titled "Alexian Brothers Health Center: Contact Information". All proximity models rank these three documents lower for not having "Hospital" close to the other terms. Finally, notice that tuning parameters for SDM on WT13 using tenfold cross evaluation resulted in settings for λ_U that vary from 0.13 to 0.46 between the folds, which could indicate that cross-evaluation may not have resulted in close to optimal parameter settings being used.

The results that were presented in Sect. 6.2, were also tested on significant improvements between systems. In Table 6, each cell contains the test sets on which the models in the row label performed significantly better than the model in the column label. As a reference, "Appendix 2" contains the highest MAP score obtained by a system that participated in the TREC ad-hoc tasks, and highest StatMAP obtained by a system that participated in the Web Track ad-hoc tasks.

6.3 Robustness

In this section, the robustness of the proximity models is compared. Here, robustness of a model is defined as the number of queries that are improved rather than hurt. Sakai et al.

Table 5 Comparison of CPE and CPES to KLD, MinDist, PLM and SDM over ad-hoc Web Track collections

		KLD	MinDist	PLM	SDM	FDM	CPE	CPES
WT09	MAP	0.0334	0.0358	0.0348	0.0439	**0.0460**	0.0416	0.0425
	δ (%)		+7.1	+4.1	+31.2	+37.4	+24.5	+27.1
	p value		0.192	0.294	0.017	0.014	0.026	0.017
WT11	MAP	0.0775	0.0920	0.0884	0.1306	**0.1416**	0.1244	0.1381
	δ (%)		+18.8	+14.1	+68.7	+82.8	+60.7	+78.3
	p value		0.000	0.011	0.002	0.000	0.000	0.000
WT12	MAP	0.0418	0.0277	0.0304	0.0437	0.0481	0.0495	**0.0545**
	δ (%)		−33.7	−27.4	+4.5	+14.9	+18.2	+30.4
	p value		0.042	0.048	0.342	0.077	0.001	0.000
WT13	MAP	0.1332	0.1387	0.1344	0.1483	0.1441	0.1556	**0.1748**
	δ (%)		+4.1	+0.9	+11.3	+8.2	+16.8	+31.2
	p value		0.002	0.146	0.003	0.143	0.001	0.000
Average	δ (%)		+2.9	+0.7	+28.2	+32.8	+29.8	+43.4

The percentage represent improvement in MAP over the KLD baseline. These improvements were tested for significance with a paired Student's t-test, one tailed, $\alpha = 0.05$. The highest MAP per test set is in bold

(2005) introduced the Reliability of Improvement metric (RI), which was later called Robustness Index by Collins-Thompson and Callan (2007). In Eq. (23), the robustness index RI is a value between −1 and 1, computed as the difference between the number of queries improved over the KLD baseline n^+ and the number of queries scoring lower than the KLD baseline n^- divided by the total number of queries $|Q|$. The comparison of the robustness indices in Table 7 shows that CPE is, on average, the most robust model.

$$RI = \frac{n^+ - n^-}{|Q|} \tag{23}$$

6.4 Proximity hypothesis

We hypothesized that the usefulness of term proximity for ranking documents is not bound to a maximum word distance. To test this, we retrieved results with the CPE model on the test collections, varying the maximum span in the range $10, 20, \ldots, 1{,}000$. In Fig. 4, the results are shown for the TREC sets and the Web Track sets, with a MAP score that was normalized by dividing by the highest MAP for the sets, so that 1 is the maximum value. For Web collections, having no constraint on the distance used maximizes results. The results for the TREC sets reaches is optimal when using co-occurrences within 140 words, and stabilizes to a level approximately 0.1 % lower over unlimited distance. Inspection revealed that this decline beyond 140 words is mostly caused by documents from the AP and WSJ sections of the TREC collections, therefore it may be a collection specific characteristic. The expected gain of finding an optimum maximum span opposed to simply using all co-occurrences will be minimal, if any. We therefore conclude that using all co-occurrences despite of their distance in text is a good principle, provided that the score of co-occurrences correctly depends on their distance in text.

Table 6 Comparison of significant improvements between six proximity baselines

Row > column	MinDist	PLM	SDM	FDM	CPE	CPES
MinDist		9 11 13				3
PLM	5 6		6			
SDM	9 11 12 13	3 9 11 12 13				3
FDM	5 9 11 12	3 9 11 12	5 11		3 9 11	3
CPE	5 9 11 12 13	8 9 11 12 13	8	8 13		3
CPES	5 8 9 11 12 13	8 9 11 12 13	8 12 13	8 13	11 12 13	

The collection numbers 1–8 for TREC 1–8 and 9–13 for Web Track 2009–2013 indicate that the proximity model in the row label significantly performed better than the proximity model in the column label. Improvements were tested for significance with a paired Student's *t*-test, one tailed, $\alpha = 0.05$

Table 7 Robustness Index (RI) of CPE and CPES to KLD, MinDist, PLM and SDM compared to the KLD baseline over ad-hoc collections

	MinDist	PLM	SDM	FDM	CPE	CPES
TREC 1	0.18	**0.22**	0.14	0.08	0.18	0.12
TREC 2	0.16	0.10	0.08	0.16	**0.20**	0.16
TREC 3	0.12	0.28	**0.32**	0.28	0.22	0.10
TREC 5	0.24	**0.36**	0.16	0.24	0.34	0.30
TREC 6	0.06	**0.26**	0.14	0.10	0.16	0.12
TREC 7	**0.28**	0.06	0.06	0.08	0.18	0.18
TREC 8	0.00	−0.04	0.12	0.20	**0.24**	**0.24**
WT09	0.04	0.00	**0.33**	0.24	0.18	0.24
WT11	0.40	0.12	0.48	0.52	0.50	**0.60**
WT12	−0.20	−0.14	0.14	0.40	0.48	**0.50**
WT13	0.16	−0.08	0.48	0.48	**0.56**	0.54
Average	0.13	0.10	0.22	0.25	0.29	0.28

In bold is the highest RI per test set

6.5 Qualitative analysis

The results were analyzed in more detail to gain insight into how the distance between occurring query terms affects the document scores, what caused relevant documents to be negatively affected, and how term combinations that have an inversely proportional relation to being relevant affect the ranking. The intended effect of a proximity model is illustrated using two cases of documents being re-ranked. On topic 365 of TREC-7: "El Nino", KLD resulted in 0.73 on average precision, ranking the irrelevant document FBIS4-11397 in 10th position. CPE ranked this specific document to 43rd position, resulting in an almost perfect average precision of 0.96. The fragment (from document FBIS4-11397) shows the part of this irrelevant document where "El" and "Nino" co-occur further apart, using the words "El" and "Nino" in a different meaning.

> Four other members of the gang that killed General Julio Nino Rios, former director of the defunct Peruvian Republican Guard. (Lima EXPRESO in Spanish 4 Apr 94 p A17). Seven Tupac Amaru Revolutionary Movement, MRTA, 'terrorists' were

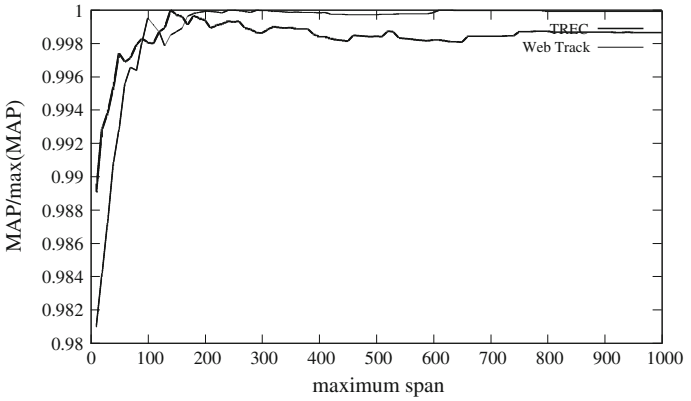

Fig. 4 A comparison of MAP obtained for the TREC sets and Web Track sets, with a varying limit on maximum span used to match occurrences. The MAP scored per maximum span was divided by maximum MAP to normalize scores to a maximum of 1

> presented to the press at the Fifth Infantry Division unit in El Milagro District, in the Amazon province of Bagua.

For TREC-7 topic 397: "Automobile Recalls", CPE improved average precision from 0.32 to 0.40. The next fragment (from document LA021390-0016), which was ranked #1 by KLD, discusses *safety seats* being recalled, not automobiles. CPE re-ranked this document to #4, below three relevant documents. Both fragments illustrate that if more intermediate words are present, it becomes more likely that the relation between co-occurring query terms is diverted or changed.

> Parents often breathe a sigh of relief once they buckle their toddlers into automobile safety seats. That might be a false sense of security. Since 1968, there have been 39 recalls of various models of child safety seats.

Queries that were negatively affected by the proximity model were also inspected. Among the worst performing queries was TREC 1 topic 61: "Israeli Role in Iran-Contra Affair". For some queries, the score of one vital term, in this case "Israeli", cannot compete with the much higher proximity score of the frequently adjacently occurring "Iran Contra Affair". Another bad case for proximity models is TREC 7 topic 377: "Cigar Smoking", for which the distance between terms is inversely related to relevance. In this case, it is partly a side effect of stemming that causes smoke, smoker and smoking to be converted to the same stem, thus promoting documents containing "cigar smoke" or "cigar smoker". Specifically inspecting term combinations with an inverse distance-relevance relationship, it appears that these occur mostly in queries with more than two terms. The last example is TREC 6 topic 350: "Health and Computer Terminals", which requests information about the hazardous effects to individuals who make daily use of computer terminals. Adjacent occurrences of "computer health" and "health terminals" all appear in non-relevant documents. For this topic only one of these six documents was promoted into the top-1000 documents by CPE, specifically to position 429. The occurrences with an opposite proximity-relevance relationship are possibly restrained by the absence of other query terms which are less likely to co-occur in irrelevant documents, reducing the negative impact of these "opposites". Although this side-effect was not studied thoroughly, on this particular topic CPE improved average precision by approx. 20 % over the KLD

baseline, illustrating that impact of irrelevant adjacent co-occurrences can be conditioned by the absence of other terms.

7 Conclusion

We studied the use of term proximity information for ranking documents according to their estimated relevance, focusing on the question how the distance between co-occurring query terms may influence relevance weights associated to these terms. We first analyzed how the distance between query terms in documents is related to their likelihood to appear in a relevant document, using data from existing test collections. The insights from this preliminary study were used to design a simple yet effective proximity model [called cumulative proximity expansions (CPEs)], that aggregates the estimated impact of query term co-occurrences. This impact is solely derived from the distance between their respective occurrences, motivated by the intuition that the distance between terms corresponds to the number of intervening terms that each may have modified the semantic relation between these terms.

CPE distinguishes itself from the state-of-the-art proximity models by requiring no additional parameters to be tuned, and no collection wide co-occurrence statistics. CPE is thus easily implemented, by modifying only the scoring function on any inverted index that contains positional posting lists for unigrams. We optimized retrieval speed by counting the co-occurrences of all term combinations in a single pass, which we show to be feasible even in the case of the very long queries of TREC-4. The runtime performance is close to that of its main competitor SDM, and more feasible in practice than FDM, should we encounter long queries.

We empirically compared the retrieval performance between our proposed models and four different state-of-the-art baseline proximity models, over seven TREC ad-hoc and four TREC Web Track collections. Of all models, the CPE model is the most robust (in terms of the Robustness Index), while, on average, retrieval effectiveness is comparable to FDM and outperforms MinDist, PLM and SDM. The CPES variant uses a query's stop words in the term combinations considered, and outperforms the other models on the Web collections. This variant is however less robust on the ad-hoc collections, where the stop words in queries do not always help predict the documents that are most relevant.

In previous research, the use of proximity information is often limited to selected term combinations and within a limited word distance. We hypothesized that the likelihood for query terms to co-occur in a relevant document diminishes with the distance between these terms, but that the evidence from co-occurrence should not be restricted to a short window size only. Our experimental results indeed confirm the results of (a large number of) previous studies, that nearby co-occurrences of query terms provide strong evidence about a document's relevance. Scoring distant co-occurrences does however lead to further improvements in effectiveness. Although we have observed that proximity can be counter-effective in special cases of query term combinations, we have found that, generally, using all term combinations outperforms other models and provides more robust results. Nevertheless, an interesting future direction of research would analyze for which queries or term combinations proximity models are likely to improve results. More insights to predict the important term combinations (as well as those to ignore) may alleviate the negative effects of proximity as well as improve the runtime performance of the system.

Appendix 1: Proof of rank equivalence KLD and QL

The KLD function presented by Zhai and Lafferty (2004) and Eq. (1) and the Query Likelihood function presented by Metzler and Croft (2005) and Eq. (10) are both Dirichlet smoothed language model estimates. Although these functions assign different scores to documents, they are in fact rank equivalent, as we will show here. In this proof, q_i is a term in query Q, μ is the Dirichlet prior parameter, $|D|$ is the number of words in document $D, tf_{q_i,D}$ is the number of times q_i appears in D, cf_{q_i} is the number of times q_i appears in collection $C, |C|$ is the total number of words in C, and $P(q_i|C)$ is the simple likelihood estimate that q_i appears in C.

$$QL(Q,D) = \sum_{q_i \in Q} \log\left(\left(1 - \frac{\mu}{\mu + |D|}\right) \cdot \frac{tf_{q_i,D}}{|D|} + \frac{\mu}{\mu + |D|} \cdot \frac{cf_{q_i}}{|C|}\right) \tag{24}$$

$$= \sum_{q_i \in Q} \log\left(\frac{\cancel{|D|}}{\mu + |D|} \cdot \frac{tf_{q_i,D}}{\cancel{|D|}} + \frac{\mu}{\mu + |D|} \cdot \frac{cf_{q_i}}{|C|}\right) \tag{25}$$

$$= \sum_{q_i \in Q} \log\left(\frac{1}{\mu + |D|} \cdot \left(tf_{q_i,D} + \frac{\mu \cdot cf_{q_i}}{|C|}\right)\right) \tag{26}$$

$$= \sum_{q_i \in Q} \log\left(\frac{1}{\mu + |D|} \cdot \frac{1}{|C|} \cdot \left(tf_{q_i,D} \cdot |C| + \mu \cdot cf_{q_i}\right)\right) \tag{27}$$

$$= \sum_{q_i \in Q} \log\left(\frac{\mu}{\mu + |D|} \cdot \frac{cf_{q_i}}{|C|} \cdot \left(1 + \frac{tf_{q_i,D} \cdot |C|}{\mu \cdot cf_{q_i}}\right)\right) \tag{28}$$

$$\underset{=}{\text{rank}} \sum_{q_i \in Q} \log\left(\frac{\mu}{\mu + |D|} \cdot \left(1 + \frac{tf_{q_i,D} \cdot |C|}{\mu \cdot cf_{q_i}}\right)\right) \tag{29}$$

$$\underset{=}{\text{rank}} \sum_{q_i \in Q} \log\left(1 + \frac{tf_{q_i,D}}{\mu \cdot Pq_i|C}\right) + |Q| \cdot \log\frac{\mu}{\mu + |D|} \tag{30}$$

$$= KLD(Q,D) \tag{31}$$

In Eq. (28), we can eliminate the likelihood that the term appears in the corpus, which is the same for every document. Equation (29) is thus rank equivalent to Eq. (28). In Eq. (30), the KLD function is derived by multiplying the document prior with a document independent constant, thus completing the proof.

Appendix 2: Results of best TREC runs

In Table 8, we present the highest StatMAP obtained by any system that participated at the TREC ad-hoc tracks. For the 2011 and 2012 Web Tracks, the best scoring systems outscore the proximity models by a large margin. The best TREC systems are complete retrieval systems, that filter out spam, use Learning To Rank on a range of features. In 2009 and 2013 there is less difference between the TREC best system and the proximity models. The first year these Web collections were used, no training data was given to the participants.

Table 8 On the left, the highest MAP score obtained by any system that participated during the TREC ad-hoc task, and on the right the highest StatMAP score obtained by any system that participated for the Web Track ad-hoc tasks

Test set	MAP	Test set	StatMAP
TREC1	n/a	WT09	0.0855
TREC2	0.3144†	WT11	0.2165
TREC3	0.4012†	WT12	0.2168
TREC5	0.2466†	WT13	0.1769
TREC6	0.2876		
TREC7	0.2614		
TREC8	0.3063		

The results were obtained by automatic systems using the topic title only, except for runs marked with a † which possibly used more information (e.g. topic description)

References

Beeferman, D., Berger, A., & Lafferty, J. (1997). A model of lexical attraction and repulsion. In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the European chapter of the association for computational linguistics* (pp. 373–380). Association for computational linguistics.

Bendersky, M., & Croft, W. B. (2012). Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval* (pp. 941–950). ACM.

Bendersky, M., Metzler, D. & Croft, W. B. (2010). Learning concept importance using a weighted dependence model. In *Proceedings of the third ACM international conference on Web search and data mining* (pp. 31–40). ACM.

Büttcher, S., Clarke, C. L., & Lushman, B. (2006). Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 621–622). ACM.

Carterette, B., Pavlu, V., Kanoulas, E., Aslam, J. A., & Allan, J. (2008). Evaluation over thousands of queries. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 651–658). ACM.

Clarke, C. L., Cormack, G. V., & Tudhope, E. A. (2000). Relevance ranking for one to three term queries. *Information Processing and Management*, *36*(2), 291–311.

Collins-Thompson, K., & Callan, J. (2007). Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 303–310). ACM.

Croft, W. B., Turtle, H. R., & Lewis, D. D. (1991). The use of phrases and structured queries in information retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 32–45). ACM.

Cummins, R., & O'Riordan, C. (2009). Learning in a pairwise term-term proximity framework for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval* (pp. 251–258). ACM.

De Kretser, O. & Moffat, A. (1999). Effective document presentation with a locality-based similarity heuristic. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 113–120). ACM.

Fagan, J. (1987). Automatic phrase indexing for document retrieval. In *Proceedings of the 10th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 91–101). ACM.

Gao, J., Nie, J.-Y., Wu, G., & Cao, G. (2004). Dependence language model for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 170–177). ACM.

Hawking, D., & Thistlewaite, P. (1995). Proximity operators-so near and yet so far. In *Proceedings of the 4th text retrieval conference* (pp. 131–143).

He, B., Huang, J. X., & Zhou, X. (2011). Modeling term proximity for probabilistic information retrieval models. *Information Sciences*, *181*(14), 3017–3031.

Keen, E. M. (1991). The use of term position devices in ranked output experiments. *Journal of Documentation*, *47*(1), 1–22.

Lavrenko, V., & Croft, W. B. (2001). Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 120–127). ACM.

Liu, X., & Croft, W. B. (2002). Passage retrieval based on language models. In *Proceedings of the eleventh international conference on information and knowledge management* (pp. 375–382). ACM.

Lv, Y., & Zhai, C. (2009). Positional language models for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval* (pp. 299–306). ACM.

Metzler, D., & Croft, W. B. (2005). A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 472–479). ACM.

Metzler, D., & Croft, W. B. (2007). Latent concept expansion using markov random fields. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 311–318). ACM.

Miao, J., Huang, J. X., & Ye, Z. (2012). Proximity-based rocchio's model for pseudo relevance. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval* (pp. 535–544). ACM.

Nallapati, R., & Allan, J. (2002). Capturing term dependencies using a language model based on sentence trees. In *Proceedings of the eleventh international conference on information and knowledge management* (pp. 383–390). ACM.

Rasolofo, Y., & Savoy, J. (2003). Term proximity scoring for keyword-based retrieval systems. In *Advances in information retrieval* (pp. 207–218). Springer.

Sakai, T., Manabe, T., & Koyama, M. (2005). Flexible pseudo-relevance feedback via selective sampling. *ACM Transactions on Asian Language Information Processing (TALIP)*, *4*(2), 111–135.

Shi, L., & Nie, J.-Y. (2010). Using various term dependencies according to their utilities. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 1493–1496). ACM.

Song, F., & Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management* (pp. 316–321). ACM.

Song, R., Taylor, M. J., Wen, J.-R., Hon, H.-W., & Yu, Y. (2008). Viewing term proximity from a different perspective. In *Advances in information retrieval* (pp. 346–357). Springer.

Svore, K. M., Kanani, P. H., & Khan, N. (2010). How good is a span of terms? Exploiting proximity to improve web retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval* (pp. 154–161). ACM.

Tao, T., & Zhai, C. (2007). An exploration of proximity measures in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 295–302). ACM.

Tellex, S., Katz, B., Lin, J., Fernandes, A., & Marton, G. (2003). Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 41–47). ACM.

Van Rijsbergen, C. J. (1977). A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, *33*(2), 106–119.

Vechtomova, O., & Wang, Y. (2006). A study of the effect of term proximity on query expansion. *Journal of Information Science*, *2*(4), 324–333.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, *22*(2), 179–214.

Zhao, J., & Yun, Y. (2009). A proximity language model for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (pp. 291–298). ACM.