Available online at www.sciencedirect.com

ScienceDirect

Information Processing and Management 44 (2008) 1485–1502

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Lexical cohesion and term proximity in document ranking

Olga Vechtomova [a,*], Murat Karamuftuoglu [b]

[a] Department of Management Sciences, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3GE
[b] Department of Computer Engineering, Bilkent University, Bilkent 06800 Ankara, Turkey

Received 30 May 2007; received in revised form 29 December 2007; accepted 9 January 2008
Available online 5 March 2008

Abstract

We demonstrate effective new methods of document ranking based on lexical cohesive relationships between query terms. The proposed methods rely solely on the lexical relationships between original query terms, and do not involve query expansion or relevance feedback. Two types of lexical cohesive relationship information between query terms are used in document ranking: short-distance collocation relationship between query terms, and long-distance relationship, determined by the collocation of query terms with other words. The methods are evaluated on TREC corpora, and show improvements over baseline systems.

Keywords: Information retrieval; Lexical cohesion; Term proximity

1. Introduction

In this paper, we present two new methods that rank documents on the basis of information on lexical cohesive relationships between query terms. The methods presented are grounded on the linguistic theory that cohesion is a characteristic of all well-formed natural language texts, which is achieved by means of various lexical and grammatical resources available to natural languages. Cohesion maintains continuity between parts of text, and distinguishes it from unconnected sequences of sentences. The importance of the presented methods is that they are founded on a firm linguistics theory of text cohesion (Halliday & Hasan, 1976; Hoey, 1991, 2005), which is shown to be applicable to text retrieval by Vechtomova, Karamuftuoglu, and Robertson (2006). We provide further evidence in this paper that lexical cohesion property of natural language texts could be used to improve effectiveness of retrieval systems. Unlike various feedback-based methods, such as blind feedback or local context analysis, the presented methods do not involve feedback or query expansion, and rely solely on the lexical cohesion information inherent in text.

The work described in this paper extends the research reported by Vechtomova et al. (2006) in two important ways: it combines short (proximity) and long-distance lexical cohesive relationship information in ranking

[*] Corresponding author. Tel.: +1 519 888 4567x32675; fax: +1 519 746 7252.
E-mail addresses: ovechtom@uwaterloo.ca (O. Vechtomova), hmk@cs.bilkent.edu.tr (M. Karamuftuoglu).

0306-4573/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2008.01.003

documents, and natural language sentence boundaries instead of arbitrary windows around query terms are used in the calculations, which is arguably a more reliable way of identifying cohesive relationships between query terms.

The results of evaluation experiments on four TREC collections we report give further support to the hypothesis put forward in Vechtomova et al. (2006) that lexical environments of distinct query terms in relevant documents are more strongly linked to each other, and thus are more cohesive, than in non-relevant documents. The paper describes in detail how lexical cohesion between query terms in documents could be used in document ranking, and points to the potential of the cohesion theory in improving effectiveness of retrieval systems.

The rest of the paper is organised as follows: in the next section we present the concept of lexical cohesion and its application in information (document) retrieval, as well as a review of term proximity and related retrieval methods; in the subsequent section, the document ranking methods we have developed on the basis of lexical cohesion analysis of documents are presented; Section 4 presents the results and analysis of the experiments conducted to evaluate the effectiveness of the developed methods; the final section summarises the experimental results and provides suggestions for future work.

2. Lexical cohesion

Cohesion is a characteristic of text, which is achieved through semantic connectedness between words in text. Halliday and Hasan (1976) suggested that semantic connectedness or cohesion of text is realised through text-forming resources of the language. They identified two major types of cohesion: (1) grammatical, realised through grammatical structures, and consisting of reference, substitution, ellipsis and conjunction; and (2) lexical, realised through lexis. Lexical cohesion, in turn, is analysed in terms of two broad categories: reiteration and collocation. Reiteration refers to a range of relations between a lexical item and another one in text, where the second lexical item can be an exact repetition of the first, a general word, its synonym or near-synonym or its superordinate. Halliday and Hasan understood collocation as a relationship between lexical items that occur in the same environment, but they did not formulate a precise definition. Collocation is used by others to refer to phrases and idiomatic expressions, whose meaning cannot be completely derived from the meaning of their elements. For example Manning and Schütze (1999) defined collocation as grammatically bound elements occurring in a certain order which are characterised by limited compositionality, i.e., the impossibility of deriving the meaning of the total from the meanings of its parts.

In this study, we investigate the relationship between document relevance and the level of lexical cohesion among query terms in a document based on two types of relationships:

- Reiteration of the words in the contexts of distinct query terms. In other words, long-span transitive collocation relationship between query terms.
- Collocation, that is proximity or short-span relationship between query terms.

The latter covers not only idiomatic expressions, but also phrasal structures that exhibit a degree of flexibility in their composition (i.e., allow intervening words, change of word order, etc.), and words related by syntactical relationships within the sentence, e.g., subject–object (cf. Section 3).

2.1. Lexical links and bonds

A single instance of a lexical cohesive relationship between two words is usually referred to as a lexical link (Ellman & Tait, 1998; Hoey, 1991; Morris & Hirst, 1991). Lexical cohesion in text is normally realised through sequences of linked words – lexical chains. The term ‘chain’ was first introduced by Halliday and Hasan (1976) to denote a relation where an element refers to another element, which in turn refers to another element and so on. Morris and Hirst (1991) define lexical chains as sequences of related words in text.

Hoey (1991) pointed that text cohesion is formed not only by links between words, but also by semantic relationships between sentences. A cohesive relation between sentences was named by Hoey as a lexical bond. A lexical bond exists between two sentences when they are connected by a certain number of lexical links. Hoey argues that an empirical method for estimating a minimum number of links the sentences should have to form a bond must rely on the proportion of sentence pairs that form bonds in text. Usually, two or three links are considered sufficient to constitute a bond between a pair of sentences. It is notable that in Hoey's experiments, only 20% of bonded sentences were adjacent pairs.

2.2. Long-span lexical cohesive relationships in IR

Long-span lexical cohesive relationships remain a relatively unexplored topic in information retrieval. Stairmand(1997) mapped the lexical contents of documents into WordNet synsets, identifying in each document lexical clusters and lexical chains. At search time, each query term, mapped into a WordNet synset, is matched against the weighted synsets representing the documents. The results demonstrated improved performance for some queries only.

Ellman and Tait (1998) applied lexical chains to re-rank Web pages retrieved by a commercial search engine. Both the Web pages and the exemplar text used to retrieve them were represented by lexical chains. Each Web page retrieved was then compared to the exemplar text taking into account the strength of every link in every chain used in representing the texts. The documents were then re-ranked based on their similarity to the query (exemplar text). Although inconclusive, the evaluation experiments suggested that there is some benefit in ranking documents in this way.

A detailed analysis of the use of lexical cohesion in IR is reported by Vechtomova et al. (2006). They hypothesised that in a relevant document all query terms are likely to be used in related contexts, which tend to share many semantically-related words. In a non-relevant document, query terms are less likely to occur in related contexts, and hence they co-occur with fewer common words. It is, therefore, hypothesised that relevant documents tend to have a higher level of lexical cohesion between different query terms' contexts than non-relevant documents. It is experimentally demonstrated in the same work that this hypothesis holds true. The same work also reports a document ranking method based on the above arguments. The results of the evaluation experiments with TREC collections demonstrated that the lexical cohesion-based method performed better than a baseline IR system.

2.3. Term proximity-based methods in IR

A wide range of document ranking methods that use term proximity have been developed. They are based on the following two intuitions: (1) the closer the terms are in a document, the more likely it is that they are related, and (2) the closer the query terms are in a document, the more likely it is that the document is relevant to the query. Some of these methods attempt to capture phrases or multi-word units in text (Fagan, 1989; Hull et al., 1997; Mitra, Buckley, Singhal, & Cardie, 1997), while others rank documents by proximity between query terms within certain text units, e.g., windows of varying sizes, sentences, paragraphs or even entire documents (Büttcher, Clarke, & Lushman, 2006; Clarke, Cormack, & Tudhope, 2000; Rasolofo & Savoy, 2003; Tao & Zhai, 2007).

A phrase is a general term referring to a wide variety of lexical associations with various degrees of idiomaticity or compositionality, such as proper nouns ('Nelson Mandela', 'United Nations'), nominal compounds ('amusement park', 'free kick', 'animal protection') and phrasal verbs ('reach out', 'sign in'). Phrases received much attention in information retrieval research. This interest can be partially attributed to the fact that phrases tend to have a higher information content and specificity than single words, and therefore represent the concepts expressed in text more accurately. Many leading statistical IR models, such as probabilistic (Spärck Jones, Walker, & Robertson, 1998) and vector-space (Salton, 1971), rely on the use of single terms and are based on strong term independence assumptions. Experimentally these models have consistently demonstrated high performance results with a variety of large test collections in the evaluation exercises such as TREC (Voorhees & Buckland, 2004). Nevertheless, many attempts have been made to introduce phrases into the retrieval process, but so far with mixed and inconclusive results.

One of the most comprehensive early evaluations of phrases in IR was undertaken by Fagan (1989). The main focus of his experiments was systematic evaluation of statistical phrases under different parameter settings, such as distance between their constituents and their frequency values. The evaluation results showed that performance of statistical phrases was in general similar to that of linguistically-derived (syntactic) phrases and better than performance of single terms. Fagan's experiments were later replicated by Hull et al. (1997), leading to only marginal performance gains from using syntactic phrases.

Mitra et al. (1997) conducted a large-scale evaluation of both syntactic and statistical phrases. By statistical phrases they understood contiguous bigrams of non-stopwords which occur in at least 25 documents. Syntactic phrases were defined in their experiments as specific Part-of-Speech sequences (e.g. Noun–Noun, Adjective–Noun). Their studies demonstrate that overall both statistical and syntactic phrases have very little effect on performance.

Clarke et al. (2000) proposed a technique of scoring documents based on query term proximity and density. They introduced the notion of cover, which is the shortest span of text containing instances of all query terms. Document score is calculated based on two assumptions: (1) the shorter the cover, the more likely the corresponding document is relevant, and (2) the more covers are in a document, the more likely the document is relevant. The evaluation on a TREC data set showed the effectiveness of the method. A similar technique was proposed by Hawking and Thistlewaite (1996), which also demonstrated promising results on a TREC data set.

There has also been some research directed towards modelling term dependencies within the language modelling framework. For example, Metzler and Croft(2005) proposed a method for modelling term dependencies via Markov random fields. The results showed significant improvements, particularly on large document collections.

Rasolofo and Savoy (2003) modified the BM25 weighting scheme to take into account proximity between pairs of query terms. For each possible pair of query terms they calculated term pair instance weight, which increases as the two terms occur closer to each other. The results of the evaluation on TREC ad-hoc collections show small improvements in average precision and larger improvements in precision at 5, 10 and 20 documents on some collections. Büttcher et al. (2006) proposed another method of using term proximity in conjunction with the BM25 weighting scheme, which shares some intuitions with our term proximity-based weighting method (cf. Section 3.1). Their evaluation on TREC Terabyte track collections showed that proximity-based method increased precision at 10 and 20 documents. Their experiments also showed that proximity has a greater effect on performance as the collection size gets larger.

Vechtomova (2006) proposed a method of matching and weighting phrases in documents, which specifically addressed the problem of weighting non-contiguous and incomplete phrase matches in documents. The experiments showed small improvements over a baseline system on a TREC collection.

Some query expansion (QE) techniques rely on term proximity to extract terms from top-ranked documents (blind feedback) for addition to the user's query. For example, in local context analysis (LCA) (Xu & Croft, 1996) noun groups that are collocated with query terms are extracted from the retrieved *N* top-ranked passages of fixed size, and ranked by the significance of their association with all query terms. Top-ranked noun phrases are then used in query expansion. Some other approaches that make use of delimited document parts (e.g., best passages and windows) following blind feedback include (Buckley & Waltz, 2000; Cormack, Clarke, Palmer, & Kisman, 2000; Ishikawa, Satoh, & Okumura, 1998; Strzalkowski et al., 2000).

3. Methodology

We propose new methods of using two types of lexical cohesive relationships between query terms in document ranking:

- short-distance relationship, i.e., collocation in the same grammatical-syntactic construct, which we consider as the sentence (Fig. 1) and
- long-distance relationship, determined by collocation with other words (Fig. 2).

The method of using short-distance collocation (proximity) relationship between query terms in document ranking is described in Section 3.1. We hypothesise that collocation of two different query terms in the same sentence helps predict the relevance of the document to the query. We also hypothesise that the closer the two query terms are to each other in a sentence, the more strongly they are related, and hence, the more evidence there is to the document's relevance to the query.

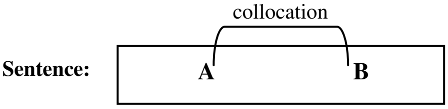

Fig. 1. Short-distance collocation (proximity) relationship between query terms A and B in the same sentence.

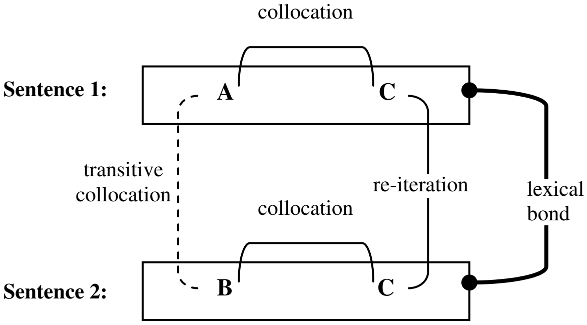

Fig. 2. Long-distance relationship between query terms A and B, determined by their collocation with term C.

The method of using long-distance relationship between query terms A and B is described in Section 3.2. If distinct query terms A and B occur in two different sentences, but both of them co-occur at least with the same *n* terms, the two query terms are considered to be related by transitive collocation, and the two sentences by a *lexical bond*. The method of document ranking we propose in Section 3.2 uses the number of lexical bonds formed between a sentence containing a query term and all other sentences that contain a different query term in calculating a document score. The above two types of lexical cohesive relationships co-exist in texts, and therefore both should be taken into account in document ranking. In Section 3.3 we propose a method for combining them.

3.1. Term proximity weighting

Our approach to proximity-based weighting of query term occurrences in the document consists of modifying the term frequency (tf) calculation in BM25. Instead of counting the actual frequency of a term's occurrence in the document to get tf, we introduce a pseudo-frequency (pf) value, calculated according to Eqs. (1) and (2). The closer the occurrence of a query term t_i is to another distinct query term's occurrence within the same sentence, the more it will contribute to the term's pf in the document. If a sentence contains only instance(s) of one query term, then each instance will contribute 1 to pf, which is equivalent to the standard tf score. The idea of using pseudo-frequency weights (Eqs. (1) and (2)) was inspired by a work on weighting terms occurring in documents with multiple fields (Robertson, Zaragoza, & Taylor, 2004), which proposes a method for weighting term frequencies based on the importance of the document field in which they occur.

$$c(t_i) = \begin{cases} 1 + \frac{1}{span(t_i,q)^p} & \text{if } q \in s;\; q \neq t_i;\; q \in Q; \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

where $c(t_i)$ – contribution of the *i* instance of the query term *t* occurring in sentence *s* to pf; span(t_i, q) – distance in number of non-stop words (stems) between the *i*th instance of the query term *t* and the nearest occurrence of any other term *q*, which belongs to the same query *(Q)*, and is not the same term as *t*; *p* – tuning constant, moderating the effect of the span size between two terms. The following parameters of *p* were evaluated (see Section 4): 0.1, 0.25, 0.5, 0.75, 1.

$$\mathrm{pf}_t = \sum_{i=1}^{N} c(t_i) \tag{2}$$

where *N* – the number of instances of query term *t* in the document.

After pf is calculated for a query term, its Term Weight (TW) in the document is calculated in the same way as in the BM25 formula (Spärck Jones et al., 1998), with pf used instead of tf (Eq. (3)):

$$\mathrm{TW}_t = \frac{(k_1 + 1) \times \mathrm{pf}_t}{k_1 \times \mathrm{NF} + \mathrm{pf}_t} \times \mathrm{idf}_t \tag{3}$$

where k_1 is the term frequency normalisation factor, which moderates the contribution of the weight of frequent terms. If $k_1 = 0$, pf has no effect on the term weight, while the higher the value of k_1 the more effect pf has on the term weight. In the evaluation described in Section 4 the following values of k_1 were evaluated: 0, 0.25, 0.5, 0.75, 1, 1.2, 1.5, 2, 2.5. NF is the document length normalisation factor, and is calculated in the same way as in the BM25 document ranking function, as expressed in Eq. (4).

$$\mathrm{NF} = (1 - b) + b \times \frac{\mathrm{DL}}{\mathrm{AVDL}} \tag{4}$$

where b is a tuning constant, DL is the document length in word counts; AVDL is the average document length in the document collection. In the evaluation described in Section 4 the following values of b were evaluated: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.75, 1.

The document matching score is calculated in the standard way as the sum of the weights of all query terms found in the document (Eq. (5)).

$$\mathrm{MS} = \sum_{t=1}^{|Q|} \mathrm{TW}_t \tag{5}$$

where $|Q|$ is the number of terms in the query.

3.2. Lexical bonds in document retrieval

Hoey (1991) stated that existence of a lexical bond between two sentences in a document suggests that they discuss the same topic. We hypothesise that by calculating lexical bonds between sentences containing different query terms, we can determine whether these terms are used in related contexts, and hence whether they discuss the same topic. For instance, consider a query from the HARD track of TREC 2005 “human smuggling”: a matching non-relevant document may discuss two independent topics such as “smuggling of drugs” and “human rights violations” in different contexts. While it is possible in the above example that sentences containing the query terms may have some words in common (such as discourse-forming words) that form lexical bonds, we hypothesise that there will be fewer bonds between these sentences compared to a relevant document which discusses “human smuggling”, mentioning both words (together or separately) several times throughout the text in related contexts. Our method aims to reward documents of the second type: i.e., those that contain different query terms in related contexts.

Initially, for each sentence s containing a query term, we calculate the number of lexical bonds formed between s and other sentences containing different query terms. More formally the number of bonds for sentence s is calculated as follows:

```
For each sentence $r \in$ Document $d$ (where $r \neq s$)
    For each query term $q_a \in s$
        For each query term $q_b \in r$
            If $q_a \neq q_b$
                Sentences $s$ and $r$ have distinct query terms
            End If
        End For
    End For

    If sentences $s$ and $r$ have distinct query terms
        Identify the number of lexical links ($links_{sr}$) between sentences $s$ and $r$
```

```
      If $links_{sr}$> $BondThreshold$
         $Bonds_s = Bonds_s + 1;$
      End If
   End If
End For
```

In the reported experiments we only considered lexical links formed by simple lexical repetition, i.e., a lexical link is considered to exist between two instances of the same lexeme, but with possible morphological variations, such as past tense forms of verbs, plural forms of nouns, etc. This is done by stemming the document representation in advance and calculating lexical links using stemmed terms.[1] It is possible to extend this method to consider other types of lexical links, e.g., formed by synonymy and hyponymy, however, experiments by Vechtomova et al. (2006) showed that lexical links formed using WordNet relationships did not yield significant improvement over links formed by simple lexical repetition in a document ranking task. This, however, could be due to the limitations of WordNet.

BondThreshold in the above algorithm is the number of links that must exist between two sentences in order for them to form a bond. We experimented with different values (1–3), with $BondThreshold = 1$ giving the best results.

In computing lexical bonds score, we follow the same principle of calculating the pseudo-frequency weights as applied to term proximity weighting (Section 3.1). After the number of bonds between sentence s and other sentences containing different query terms is determined, the contribution to pseudo-frequency of the i instance of query term t occurring in sentence s is calculated as follows:

$$c(t_i) = 1 + n \times \frac{Bonds(s)}{AveBonds} \tag{6}$$

where $Bonds(s)$ – number of bonds sentence s has with other sentences containing different query terms; n – normalisation factor ($0 \leqslant n \leqslant 1$); $AveBonds$ – average number of bonds between sentences in the document, calculated as follows:

$$AveBonds = \frac{TotalBonds}{NumSent} \tag{7}$$

where $TotalBonds$ – total number of lexical bonds formed between all sentences in the document; $NumSent$ – total number of sentences in the document.

The pseudo-frequency weight (pf), term weight (TW) and document matching score (MS) are then calculated in the same way as described in Section 3.1 above.

3.3. Combining the proximity and bond scores

In addition to ranking documents based on lexical bonds and term proximity separately, we propose to use the combination of the two factors in calculating the pseudo-frequency weight:

$$c(t_i) = \begin{cases} 1 + n \times \frac{Bonds(S)}{AveBonds} + \frac{1}{span(t_i,q)^p} & \text{if } q \in s;\ q \neq t_i;\ q \in Q; \\ 1 + n \times \frac{Bonds(S)}{AveBonds} & \text{otherwise.} \end{cases} \tag{8}$$

The pseudo-frequency weight (pf), term weight (TW) and document matching score (MS) are then calculated in the same way as described in Section 3.1 above. In the following section, we highlight differences between the methods presented here and two related methods previously developed.

3.4. Comparison with phrase-based and lexical links-based document ranking

In Vechtomova (2006), a proximity-based ranking method is described. In this method all terms in the "Title" field of a TREC topic are treated as a single phrase (referred to henceforth as query phrase). For the query phrase, all possible contiguous and non-contiguous subphrases, including the original query phrase,

[1] Porter's stemmer was used for this purpose (Porter, 1980).

are recorded in a list ranked in descending order of their length. For each subphrase in the list, we extract the minimal matching strings[2] in the document containing all the words of the subphrase in any order. Each time a minimal matching string is found, it is recorded and removed from the document representation, and the procedure is repeated with the same subphrase until no matching string is found, in which case the program attempts to match the next subphrase in the list, and so on. Matching strings containing query phrases are referred to as windows. The sets of windows in the document containing exactly the same phrase words, but possibly within different spans and in different order, are grouped into bins. All windows in each bin receive the same weight *BinWindowWeight*, which is calculated as the sum of *idf* values of all words constituting the query phrase instance in the window. Matching Score for a document, MS, is calculated as

$$\mathrm{MS} = \sum_{n=1}^{|\mathrm{bin}|} \left(\frac{(k_1 + 1) \times \mathrm{wf}_n}{k_1 \times \mathrm{NF} + \mathrm{wf}_n} \times \mathit{BinWindowWeight}_n \right) \tag{9}$$

where wf_n is the window frequency in the bin n (Eq. (10)); $\mathit{BinWindowWeight}_n$ is the weight of the windows in bin n; k_1 is the window frequency normalisation factor, which moderates the contribution of the weight of frequent windows; *NF* is the document length normalisation factor (calculated in the same way as in the BM25 document ranking function (Eq. (4))).

$$wf = \sum_{w=1}^{|window|} \frac{1}{span_w^p} \tag{10}$$

where $span = pos(l) - pos(f)$, $pos(l)$ – position number of the last query term in the window w, and $pos(f)$ – position number of the first query term in the window w; p is a tuning parameter to adjust the effect of span on wf.

The new method proposed in this paper (Section 3.2) has the following advantages over the above method:

- In the new method, we only count proximity between distinct query terms within the same sentence. In the old method, the entire document constitutes the context within which term proximity is calculated. Arguably, within-sentence co-occurrence is a more reliable indication of terms' relatedness than within-document co-occurrence.
- In the old method there is a problem of inconsistent window frequency computation, as the same query term may be part of different subphrases and therefore allocated to different bins. To illustrate the problem, consider the query "Hubble Telescope Achievement". Two of the bins could be "Hubble Telescope" and "Telescope Achievement" with corresponding window frequencies (wf) of 3 and 3 in document A, and 5 and 1 in document B. Due to the non-linear window frequency normalisation factor (*k1*), each window containing "Hubble Telescope" in document B would contribute less to wf than each window with "Hubble Telescope" in document A. This problem is avoided in the new method as the pseudo-frequency (pf) in Eq. (2) is calculated for each query term based on all its occurrences in a document, rather than separately for each bin.

Vechtomova et al. (2006) present a method, which calculates lexical cohesion between the contexts of distinct query terms. The context of a query term in the document is understood as all non-stopwords extracted from fixed-size windows surrounding every instance of the term in the document. The number of lexical links is counted between the contexts of each pair of distinct query terms, and is normalised to give the document's lexical cohesion score (LCS_{links}) as follows:

$$\mathrm{LCS}_{links} = \frac{L}{V} \tag{11}$$

where L – the total number of lexical links in a document; V – the size (in non-stopwords) of all merged windows in a document.

The document matching score (COMB-LCS) is calculated by linearly combining BM25 document matching score with LCS as follows:

[2] Minimal matching string (MMS) is a stretch of text which contains all terms in a subphrase. Each MMS may contain only one instance of each of the terms in the subphrase. MMSs are extracted using *cgrep* (Clarke & Cormack, 1995).

Table 1
Comparison of the proposed methods with the previously developed methods (HARD 2004)

Run name	MAP	P10	R-Prec	Bpref
BM25 Wumpus ($b = 0.1$; $k_1 = 1.5$)	0.2222	0.3622	0.2685	0.2413
Lexical links (Vechtomova et al., 2006) $x = 3$	0.2346	0.3578	0.2754	0.2511
Proximity (Vechtomova, 2006) $k_1 = 2.5$; $p = 0.2$	0.2307	0.3600	0.2712	0.2531
Bonds ($n = 0.25$; $k_1 = 1.2$; $b = 0.1$)	0.2360	0.3711	0.2748	0.2603
Proximity ($p = 0.5$; $k_1 = 0.75$; $b = 0.1$)	0.2362	0.3911	0.2769	0.2621
Combined ($n = 0.5$; $p = 0.75$; $k_1 = 1.2$; $b = 0.1$)	0.2401	0.3889	0.2827	0.2638

Table 2
Comparison of the proposed methods with the previously developed methods (HARD 2005)

Run name	MAP	P10	R-Prec	Bpref
BM25 Wumpus ($b = 0.3$; $k_1 = 0.75$)	0.1984	0.4560	0.2596	0.2415
Lexical links (Vechtomova et al., 2006) $x = 0.25$	0.2092	0.4460	0.2624	0.2448
Proximity (Vechtomova, 2006) $k_1 = 2$; $p = 0.3$	0.1752	0.3760	0.2229	0.2228
Bonds ($n = 1$; $k_1 = 1$; $b = 0.1$)	0.2006	0.4580	0.2592	0.2448
Proximity ($p = 0.75$; $k_1 = 0.75$; $b = 0.2$)	0.2012	0.4560	0.2633	0.2481
Combined ($n = 1$; $p = 1$; $k_1 = 1.5$; $b = 0.1$)	0.2126	0.4700	0.2739	0.2554

$$\mathrm{COMB\text{-}LCS} = \mathrm{MS}_{\mathrm{BM25}} + x \times \mathrm{LCS}_{links} \tag{12}$$

where x is a tuning constant which regulates the effect of LCS.

The method proposed in this paper (Section 3.1) has three important improvements over the old method presented above:

- Lexical cohesion between distinct query terms is estimated by calculating lexical bonds between sentences containing distinct query terms, instead of arbitrary windows.
- The old method has a disadvantage of random attribution of collocates to query terms: if the windows of two distinct query terms a and b overlap, we cannot attribute the words in these windows to both terms, because in this case each word would form a link with itself and artificially boost the link count. It is often not possible to determine whether collocates in the overlapping parts of two windows belong to a or b, therefore they are attributed to one of them randomly. In the new method all words occurring in the sentence containing both query terms a and b are considered as collocates of both, however, this does not pose a problem because we do not calculate the bond of the sentence with itself.
- Improved normalisation. The number of lexical bonds formed between sentence s containing a query term and other sentences containing different query terms is normalised by the average number of bonds formed between any sentences in the document. The rationale is that we should not reward documents which have higher overall lexical cohesion, but only those documents, whose query-containing sentences are more cohesive compared to the overall document.

To compare the new proximity- and bonds-based methods proposed in this paper to the methods presented in Vechtomova (2006) and Vechtomova et al. (2006), we have tested them on HARD 2004 and HARD 2005 collections. The results given in Tables 1 and 2 show that the new methods outperform the old ones in both collections in most measures. The b and k_1 parameters of BM25 runs shown in the tables are those yielding the highest performance in Precision at 10 retrieved documents (P10) in the corresponding collections. The details of the evaluation experiments are reported in the next section.

4. Evaluation and discussion of results

In this section we present the evaluation of the methods proposed in this paper. Evaluation was conducted using the data from four TREC collections summarised in Table 3.

Table 3
Collection statistics

Collection	Number of topics	Number of documents
HARD 2003 (no gov. docs)	50	321,405
Robust 2004 (no gov. docs)	250	474,341
HARD 2004	50	635,650
HARD 2005	50	1,036,805

Table 4
b and k_1 values in BM25 and BM25tp giving highest performance in MAP

Collection	BM25 (Wumpus)		BM25tp	
	b	k_1	b	k_1
HARD 2003 (no gov. docs)	0.6	1.5	0.4	2
Robust 2004 (no gov. docs)	0.3	0.75	0.3	0.75
HARD 2004	0.3	1	0.4	2
HARD 2005	0.2	1.5	0.2	1.5

Table 5
b and k_1 values in BM25 and BM25tp giving highest performance in P10

Collection	BM25 (Wumpus)		BM25tp	
	b	k_1	b	k_1
HARD 2003 (no gov. docs)	0.3	2	0.5	2
Robust 2004 (no gov. docs)	0.3	1	0.3	1
HARD 2004	0.1	1.5	0.1	0.5
HARD 2005	0.3	0.75	0.2	0.5

One of the baselines used in our evaluation experiments was the well-known BM25 document ranking function implemented in the Wumpus IR system[3] (Büttcher et al., 2006). Before commencing the experiments, we evaluated BM25 with different values for k_1 and b (default values are 1.2 and 0.75, respectively) on all four collections. Values giving the highest performance in Mean Average Precision (MAP) and P10 are summarised in Tables 4 and 5. As it is seen in these tables, different values of the parameters k_1 and b, yielded the best performance in different collections. Therefore, we compared the experimental runs to the BM25 run with the best performing b and k_1 values in each collection.

As a second baseline we have used the method (BM25tp) reported in (Büttcher et al., 2006) and also implemented in Wumpus. We have selected this method because it is a proximity-based method, hence comparable to the methods reported in this paper, and it is shown to have yielded a good performance in an experiment using the TREC collections. Similarly, we have evaluated this method in different collections, and results suggest that different values of b and k_1 yielded the best performance in different collections (Tables 4 and 5).

To construct the queries, all terms from the Title field of TREC topics were used in all runs. "Proximity" runs given in Table 6 were conducted by using the proximity method (Section 3.1). "Bonds" runs shown used the method described in Section 3.2, and "Combined" runs used the method described in Section 3.3. Top 2000 documents were retrieved using BM25 implemented in Wumpus with the best b and k_1 parameters (given in Table 5) for each collection and re-ranked using each of the experimental methods, which were implemented as a set of Perl scripts. Because there was a slight variation in stemming and stopword use between our system and Wumpus, we also report the performance of our implementation of BM25 (referred to as BM25-u in Tables 6 and A1). "Proximity", "Bonds" and "Combined" runs were conducted using the same Perl scripts as "BM25-u", which, therefore, can be considered as a more appropriate baseline than BM25 implemented in Wumpus. Comparison to BM25-u allows us to isolate the effect of proximity and bonds methods on performance from other factors, such as stemming and stopwords.

[3] http://www.wumpus-search.org/.

Table 6
Best runs in each collection. Experimental runs marked with * and ** are statistically significant compared to BM25-u at 0.05 and 0.02 significance levels, respectively

Run	MAP	P10	R-Prec	Bpref
HARD 2004				
BM25 Wumpus ($b = 0.1$; $k_1 = 1.5$)	0.2222	0.3622	0.2685	0.2413
BM25tp ($b = 0.1$; $k_1 = 0.5$)	0.2280	0.3689	0.2670	0.2486
BM25-u ($b = 0.1$; $k_1 = 1$)	**0.2348**	**0.3689**	**0.2753**	**0.2615**
Proximity ($p = 0.5$; $k_1 = 0.75$; $b = 0.1$)	0.2362	0.3911	0.2769	0.2621
Bonds ($n = 0.25$; $k_1 = 1.2$; $b = 0.1$)	0.2360	0.3711	0.2748	0.2603
Combined ($n = 0.5$; $p = 0.75$; $k_1 = 1.2$; $b = 0.1$)	0.2401	0.3889	0.2827	0.2638
HARD 2005				
BM25 Wumpus ($b = 0.3$; $k_1 = 0.75$)	0.1984	0.4560	0.2596	0.2415
BM25tp ($b = 0.2$; $k_1 = 0.5$)	0.2163	0.4640	0.2789	0.2656
BM25-u ($b = 0.3$; $k_1 = 0.75$)	**0.1947**	**0.4420**	**0.2579**	**0.2399**
Proximity ($p = 0.75$; $k_1 = 0.75$; $b = 0.2$)	0.2012	0.4560	0.2633	0.2481
Bonds ($n = 1$; $k_1 = 1$; $b = 0.1$)	0.2006*	0.4580	0.2592	0.2448
Combined ($n = 1$; $p = 1$; $k_1 = 1.5$; $b = 0.1$)	0.2126**	0.4700	0.2739**	0.2554**
HARD 2003				
BM25 Wumpus ($b = 0.3$; $k_1 = 2$)	0.3279	0.5292	0.3459	0.3689
BM25tp ($b = 0.5$; $k_1 = 2$)	0.3430	0.5479	0.3550	0.3831
BM25-u ($b = 0.4$; $k_1 = 1.5$)	**0.3383**	**0.5771**	**0.3564**	**0.3856**
Proximity ($p = 0.75$; $k_1 = 2.5$; $b = 0.4$)	0.3430	0.5708	0.3635	0.3912
Bonds ($n = 0.25$; $k_1 = 1$; $b = 0.2$)	0.3315	0.5792	0.3475	0.3829
Combined ($n = 0.5$; $p = 1$; $k_1 = 1$; $b = 0.4$)	0.3409	0.5771	0.3599	0.3923
Robust 2004				
BM25 Wumpus ($b = 0.3$; $k_1 = 1$)	0.2646	0.4406	0.3029	0.2711
BM25tp ($b = 0.3$; $k_1 = 1$)	0.2785	0.4542	0.3141	0.2829
BM25-u ($b = 0.3$; $k_1 = 0.75$)	**0.2646**	**0.4462**	**0.3041**	**0.2732**
Proximity ($p = 0.75$; $k_1 = 1$; $b = 0.3$)	0.2751**	0.4566*	0.3112*	0.2816**
Bonds ($n = 0.5$; $k_1 = 1$; $b = 0.2$)	0.2665	0.4514	0.3015	0.2733
Combined ($n = 0.25$; $p = 0.5$; $k_1 = 1.2$; $b = 0.3$)	0.2747**	0.4635**	0.3107*	0.2811**

The standard performance measures were calculated for the top 1000 re-ranked documents. In Table A1 in Appendix we show all the runs which achieved the highest performance in MAP and P10 in at least one collection. For the purpose of comparison, we report the results of these runs in all other collections. For easy reference we present an extract from Table A1 showing only the runs which yielded the highest P10 in each collection (Table 6). The results are reported in MAP, P10, R-Precision, and Binary Preference (Bpref).

As can be seen from Table 6, many of the experimental runs improved performance over the baselines. Table 6 shows that in some collections the bonds-based, in others the proximity-based method yielded better performance. Experimental runs marked with * and ** in the table are statistically significant at 0.05 and 0.02 significance levels, respectively, compared to BM25-u. It is also worth emphasising that the methods reported in this study do not involve query expansion. Although, some of the methods found in the literature reported to have achieved higher performance in the Robust 2005 track,[4] they all expanded the original queries with terms derived either from external sources or the test collection (Voorhees, 2006).

It may be useful to look at the terms that form bonds between sentences (link-terms) in relevant and non-relevant documents. For instance, in topic 407 "Chimpanzee Language Ability" (HARD 2004), the following terms form bonds between sentences in relevant documents (excluding the query terms and in decreasing frequency of occurrence): "view, situation, primate, hear, different, challenge, animal, human". In non-relevant documents, the terms that form bonds, on the other hand, are the following: "china, england/english, microsoft, spain/spanish, speak, human, java, univision". Terms such as "primate", "animal" and "human" that

[4] The Robust track used the same test collection as the HARD track in TREC 2005.

form bonds in relevant documents have an obvious relation to the subject matter of topic 407. In non-relevant documents there seems to be a number of off-topic link terms such as “microsoft”, “java”, and ”univision”. The first two terms seem to refer to computer technology and programming languages, and the last term refers to a Spanish-language TV station in the US. As another example consider topic 428 (HARD 2004) “International organ traffickers”. The following are the link-terms in relevant and non-relevant documents, respectively: “human, south africa, kidney, brazil/brazilian, federal, durban, syndicate”, ”drug, police, federal, child/children, USA/united states, report, afghanistan”. Two of the link-terms in non-relevant documents, namely, ”drug” and “afghanistan” are most likely off-topic and related to drug rather that organ trafficking. Another interesting example is topic 651 “US ethnic population” (HARD 2005). The link-terms in relevant documents are: “population, hispanic, USA/united states, minorities, ethnic, american/america, group, black, catholic, latino”, while link-terms in non-relevant documents are: “ethnic, albanian, kosovo, USA/united states, population, minorities, serb, china, group, yugoslavia”. Clearly, terms such as “yugoslavia”, “albanian” and “kosovo” are related to the subject of ethnic minorities, but not in the US. It may be possible to improve performance of the Bonds method further if one counts only links between a subset of terms, such as those identified using the blind feedback process.

As expected in many topics link-terms in relevant and non-relevant documents seem to be on-topic, and it may be difficult to know which ones are better without looking at the actual context in which they are used. The following are the link-terms in relevant and non-relevant documents for topic 427 in HARD 2004 “Brazilian Landless Workers Movement”: “brazil/brazilian, lula, squatter, sao, paulo, reform”, ”brazil/brazilian, silva, party, USA/united states, lula, farm”.

The combination of the proximity matching score with the lexical bonds score led to noticeably better performance than either method only in HARD 2005 (3.4% in MAP, 2.6% in P10). There are smaller perfor-

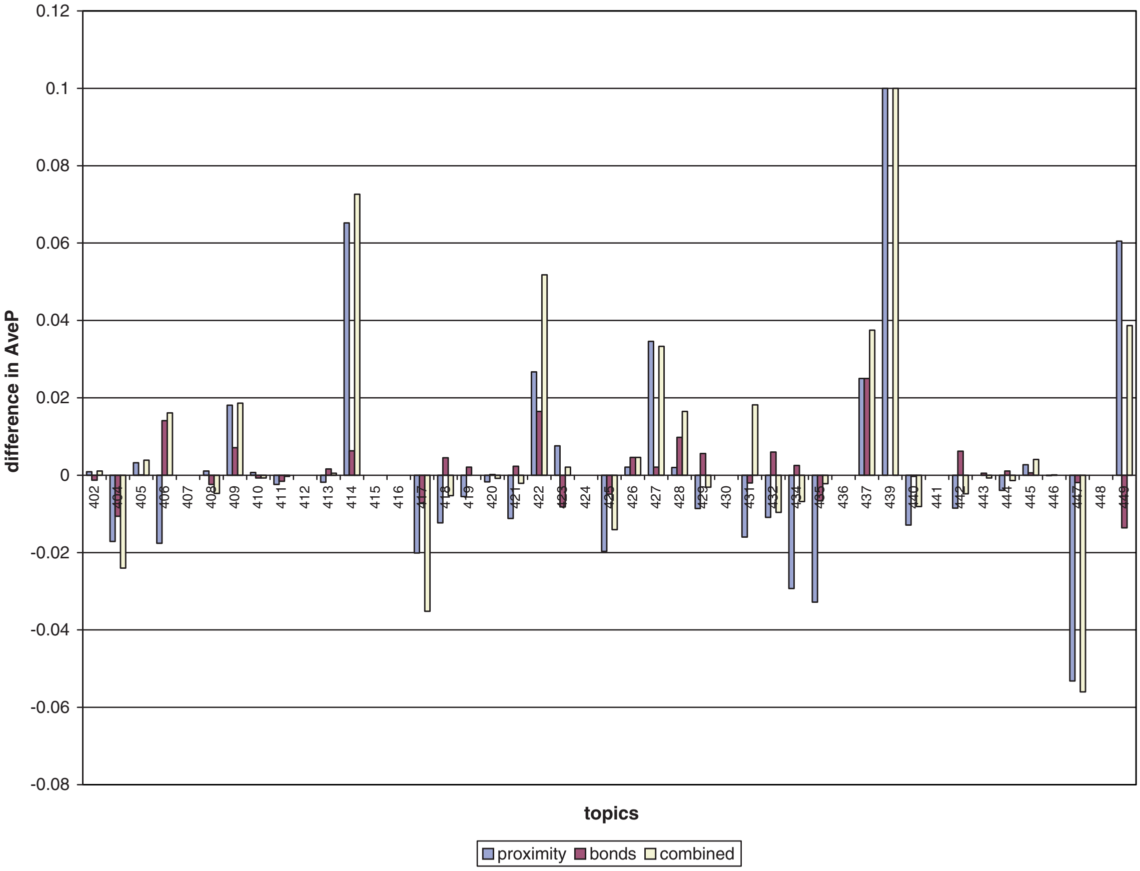

Fig. 3. Topic-by-topic difference in Average precision of the three methods from BM25-u ($b = 0.1$; $k_1 = 1$) in HARD 2004.

mance improvements in other collections in both MAP and P10. Figs. 3–6 in Appendix show the differences in Average Precision and P10 from the best baseline runs (BM25-u) of the three methods per topic in HARD 2004 and HARD 2005. The results given in these figures show that in a considerable number of cases either the proximity-, or the bonds-based methods yielded the most gain in performance. It can also be observed that only in a number of topics the combination of these two methods yielded performance gains. This suggests that there is a room for improvement in the formula given in Eq. (8) used for combining the proximity- and bonds-based methods (see discussion in Section 5).

An example of the topic which benefited from the use of the Proximity method is Topic 341 in HARD 2005, “Airport Security”, where Average Precision increased from 0.1037 in the baseline run (BM25-u) to 0.1351 in the Proximity run, but only to 0.1182 in the Bonds runs. The query is comprised of “Airport Security” a stable phrase. It can be reasoned that queries that consist of stable phrases benefit better from the Proximity method than the Bonds method. An example of the topic which was improved by the Bonds method is topic 419 in HARD 2005: “recycle, automobile tires”: Average Precision in this topic increased from 0.0863 in the baseline run (BM25-u) to 0.1012 in the Bonds run, while it dropped to 0.0806 in the Proximity run. A relevant document on this topic may not contain the words, such as “recycle” and “automobile” in the same sentence as they are not a stable idiomatic phrase. However, the method based on lexical bonds would reward those documents in which the two words occur in similar contexts in different sentences, i.e., their sentences have lexical bonds. Further research is needed to understand the effect of the two methods on different types of queries.

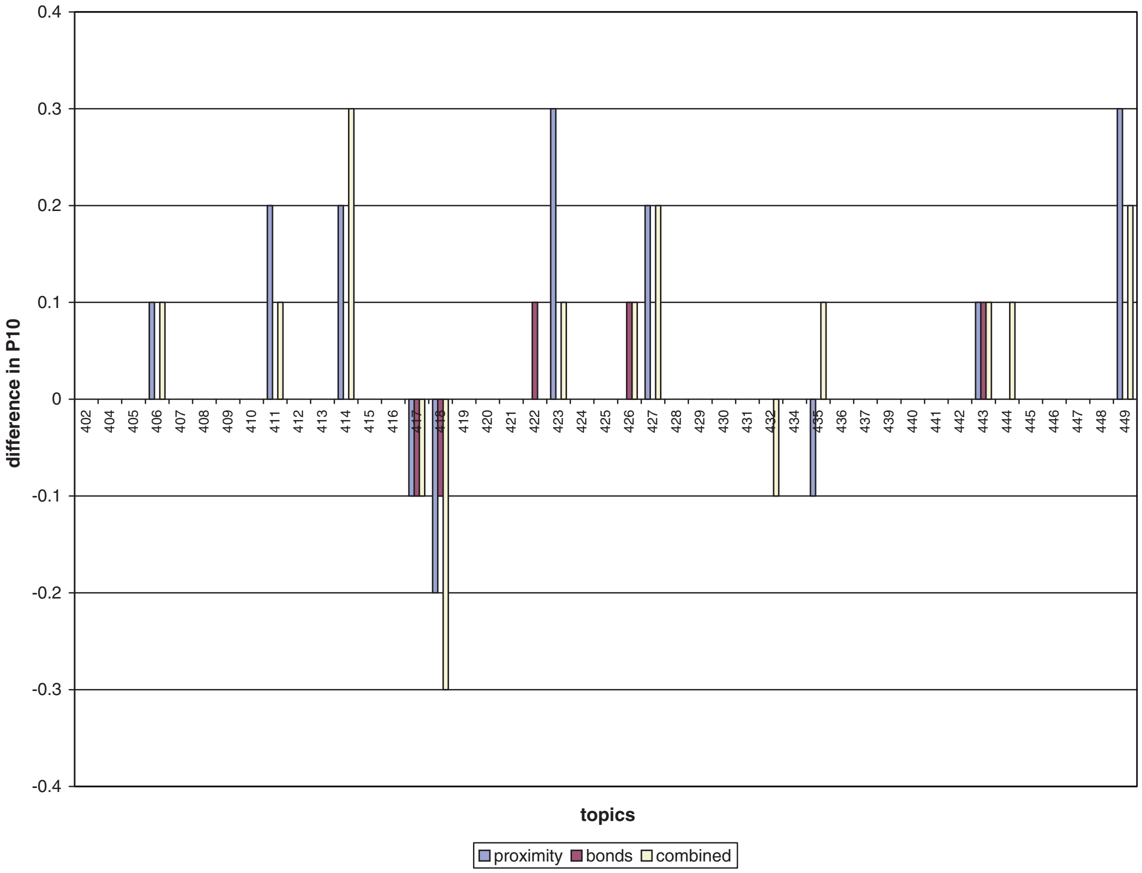

Fig. 4. Topic-by-topic difference in P10 of the three methods BM25-u ($b = 0.1$; $k_1 = 1$) in HARD 2004.

5. Conclusions and future work

In this work we investigated the effect of two types of lexical cohesive relationships on document ranking: the short-span collocation relationship (proximity) between query terms, and the long-span transitive collocation relationship (lexical bonds), established by the co-occurrence of the query terms with the same words. We developed two novel document scoring methods by considering these two types of relationships, both of which extend the well-known BM25 term weighting function. Both the proximity and lexical bonds methods showed some performance improvements over the baselines – BM25 and BM25tp. The combination of both methods achieved further gains in performance in some topics.

Our analysis shows that there is a number of topics that are improved by only one of the methods. This prompts the next line of research to determine, based on some characteristics of the query, which method is more likely to perform better with a particular query. One possible idea is to investigate the use of various collocation measures to determine how stable the co-occurrence of query terms is. It may then be possible to adjust the relative contribution of the proximity and bonds to the pseudo-frequency component of the combined document score. For example, if they are more likely to occur as a stable expression, the contribution of the Proximity score to the overall document score could be boosted. Alternatively, if they are more likely to occur separately in text, the effect of the Bonds method could be increased. For example, documents containing the query terms "amusement park" in adjacent positions are more likely to be relevant, therefore we can place more weight on the proximity factor, rather than the lexical bonds factor. On the other hand, query terms "US, investment, Africa" are not a nominal compound as "amusement park", and relevant documents may not necessarily contain these words in the same sentence. Queries

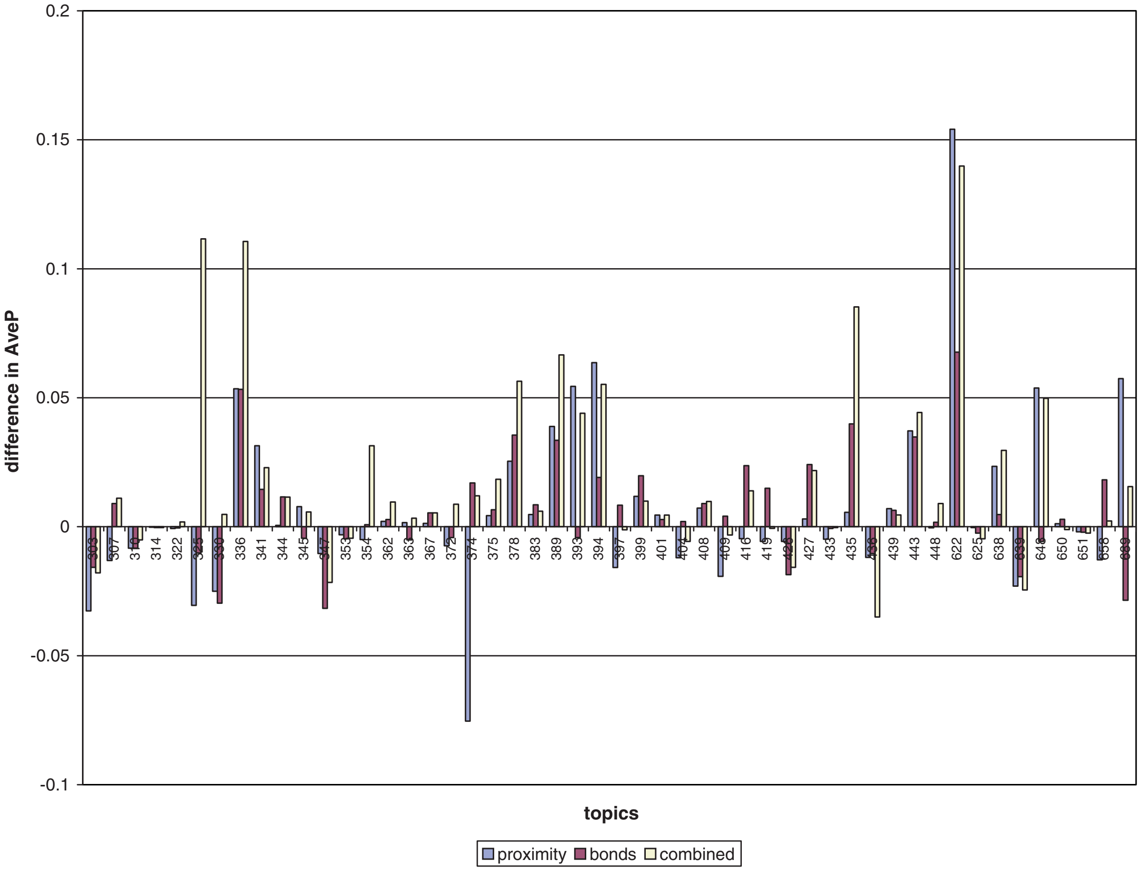

Fig. 5. Topic-by-topic difference in Average precision of the three methods.from BM25-u ($b = 0.3$; $k_1 = 0.75$) in HARD 2005.

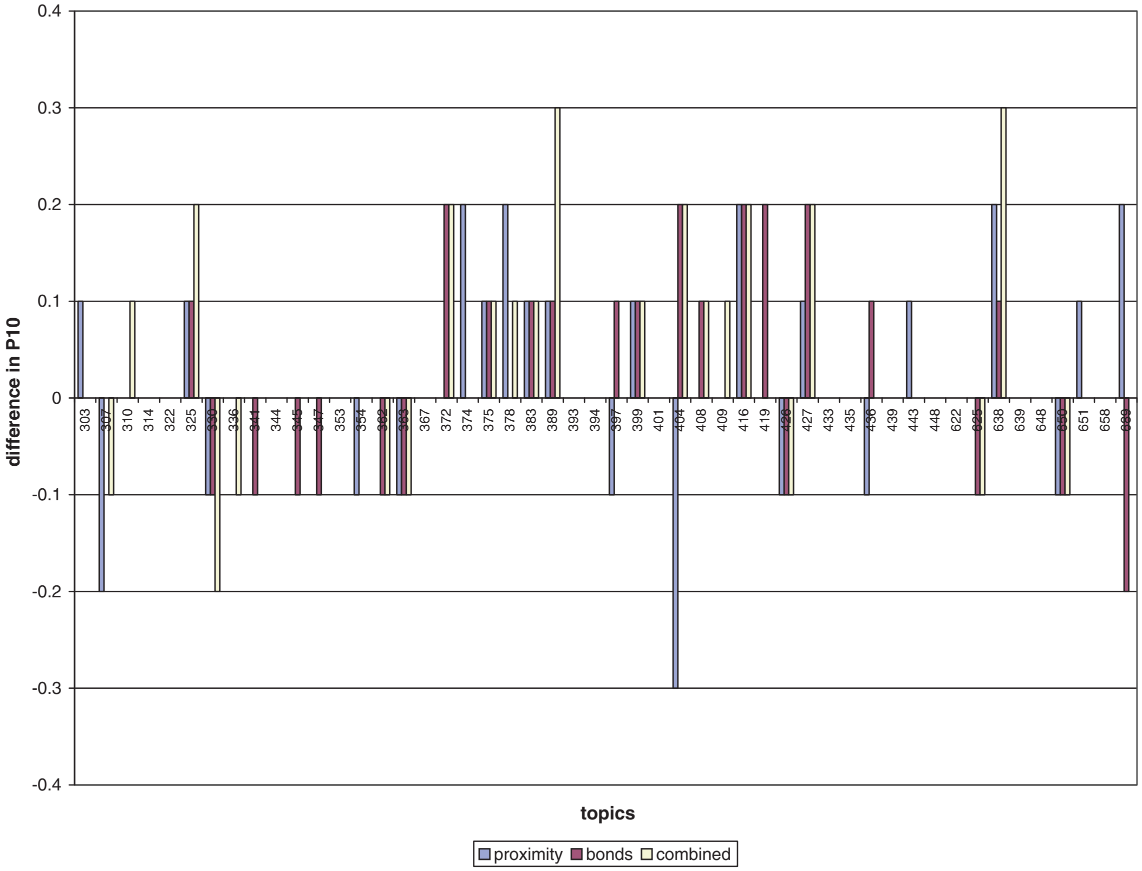

Fig. 6. Topic-by-topic difference in P10 of the three methods from BM25-u ($b = 0.3$; $k_1 = 0.75$) in HARD 2005.

of this type may benefit from increasing the weight of the lexical bonds factor in the combined document score.

Another possible way to improve the lexical cohesion model proposed in this paper is by means of query structuring, so that proximity and bonds are considered only between certain query terms. We did a preliminary analysis of some of the queries from the HARD 2005 track collection, which demonstrate promising results. Consider, for example, topic 310 ”Radio Waves and Brain Cancer”. Intuitively, one expects that bonds should be calculated between pairs of sentences that contain different phrases – either “brain cancer”, or “radio waves”. We tested this hypothesis by manually structuring the query as follows:

1. (brain AND cancer) BOND (radio AND waves)
2. (brain OR cancer) BOND (radio OR waves)

In the first query structure bonds are calculated between only those sentences, one of which contains both “brain” and “cancer” and the other “radio” and “waves”. The second structure has more relaxed constraints: one of the sentences must contain “brain” or “cancer”, while the other “radio” or “waves”. The results for this topic showed that the second structure outperforms the Bonds method reported in the paper. The main research question here is how to automatically determine the optimal query structure. We are currently investigating different methods, such as mutual information, for building such query structures automatically.

Appendix

Table A1
Performance of the experimental and baseline runs.

Run name	HARD 2004				HARD 2005				HARD 2003 (no gov.docs)				Robust 2004 (no gov.docs)			
	MAP	P10	R-prec	Bpref	MAP	P10	R-prec	Bpref	MAP	P10	R-prec	Bpref	MAP	P10	R-prec	Bpref
BM25 (implementation in Wumpus)																
$b = 0.3$; $k_1 = 1$	0.2341	0.3533	0.2712	0.2497	0.2039	0.446	0.2625	0.2418	0.3211	0.5063	0.3357	0.3616	0.2646	0.4406	0.3029	0.2711
$b = 0.2$; $k_1 = 1.5$	0.2270	0.3556	0.2709	0.2421	0.2074	0.4440	0.2695	0.2424	0.3216	0.5083	0.3413	0.3675	0.2580	0.4329	0.2946	0.2647
$b = 0.6$; $k_1 = 1.5$	0.2118	0.3400	0.2470	0.2264	0.1924	0.3800	0.2583	0.2324	0.3361	0.5125	0.3496	0.3777	0.2516	0.4281	0.2903	0.2587
$b = 0.3$; $k_1 = 0.75$	0.2333	0.3489	0.2740	0.2517	0.1984	0.4560	0.2596	0.2415	0.3116	0.5021	0.3279	0.3522	0.2656	0.4398	0.3018	0.2722
$b = 0.1$; $k_1 = 1.5$	0.2222	0.3622	0.2685	0.2413	0.2028	0.4440	0.2653	0.2401	0.3108	0.4813	0.3316	0.3638	0.2528	0.4213	0.2886	0.2595
$b = 0.3$; $k_1 = 2$	0.2182	0.3444	0.2554	0.2284	0.2049	0.4320	0.2673	0.2413	0.3279	0.5292	0.3459	0.3689	0.2503	0.4297	0.2885	0.2587
$b = 0.75$; $k_1 = 1.2$ (default)	0.2094	0.3133	0.2406	0.2250	0.1797	0.3560	0.2438	0.2244	0.3290	0.4937	0.3400	0.3753	0.2450	0.4133	0.2855	0.2542
BM25tp																
$b = 0.4$; $k_1 = 2$	0.2370	0.3533	0.2685	0.2538	0.2221	0.4220	0.2795	0.2615	0.3434	0.5417	0.3551	0.3814	0.2648	0.4426	0.2996	0.2700
$b = 0.2$; $k_1 = 1.5$	0.2353	0.3489	0.2683	0.2500	0.2297	0.4460	0.2877	0.2704	0.3334	0.5125	0.3485	0.3769	0.2735	0.4502	0.3099	0.2784
$b = 0.3$; $k_1 = 0.75$	0.2352	0.3489	0.2700	0.2567	0.2211	0.4540	0.2831	0.2679	0.3215	0.5021	0.3397	0.3646	0.2799	0.4534	0.3154	0.2847
$b = 0.1$; $k_1 = 0.5$	0.2280	0.3689	0.2670	0.2486	0.2152	0.4440	0.2746	0.2663	0.2948	0.4854	0.3125	0.3459	0.2758	0.4418	0.3074	0.2808
$b = 0.2$; $k_1 = 0.5$	0.2314	0.3622	0.2706	0.2550	0.2163	0.4640	0.2789	0.2656	0.3023	0.4854	0.3188	0.3510	0.2783	0.4490	0.3104	0.2834
$b = 0.5$; $k_1 = 2$	0.2314	0.3511	0.2642	0.2491	0.2168	0.4100	0.2804	0.2580	0.3430	0.5479	0.3550	0.3831	0.2614	0.4378	0.2978	0.2675
$b = 0.3$; $k_1 = 1$	0.2364	0.3511	0.2715	0.2563	0.2255	0.4540	0.2863	0.2691	0.3311	0.5125	0.3474	0.3742	0.2785	0.4542	0.3141	0.2829
BM25-u (our implementation of BM25)																
$b = 0.3$; $k_1 = 1$	0.2342	0.3511	0.2757	0.2546	0.1999	0.4380	0.2591	0.2414	0.3352	0.5708	0.3493	0.3829	0.2636	0.4438	0.3031	0.2717
$b = 0.1$; $k_1 = 1.2$	0.2362	0.3667	0.2757	0.2612	0.1991	0.4160	0.2592	0.2424	0.3244	0.5521	0.3436	0.3806	0.2554	0.4313	0.2907	0.2626
$b = 0.1$; $k_1 = 1$	0.2348	0.3689	0.2753	0.2615	0.1963	0.4240	0.2593	0.2412	0.3215	0.5521	0.3399	0.3790	0.2566	0.4341	0.2914	0.2634
$b = 0.2$; $k_1 = 1.5$	0.2339	0.3511	0.2670	0.2500	0.2035	0.4120	0.2622	0.2439	0.3307	0.5542	0.3505	0.3845	0.2577	0.4382	0.2915	0.2639
$b = 0.6$; $k_1 = 1.5$	0.2172	0.3378	0.2596	0.2426	0.1909	0.3800	0.2471	0.2330	0.3396	0.5625	0.3652	0.3937	0.2488	0.4309	0.2898	0.2577
$b = 0.4$; $k_1 = 1.5$	0.2244	0.3467	0.2676	0.2457	0.1999	0.4220	0.2568	0.2407	0.3383	0.5771	0.3564	0.3856	0.2573	0.4402	0.2951	0.2640
$b = 0.3$; $k_1 = 0.75$	0.2326	0.3533	0.2760	0.2549	0.1947	0.4420	0.2579	0.2399	0.3292	0.5563	0.3437	0.3810	0.2646	0.4462	0.3041	0.2732
$b = 0.1$; $k_1 = 1.5$	0.2305	0.3622	0.2753	0.2541	0.2004	0.4080	0.2593	0.2423	0.3241	0.5500	0.3441	0.3819	0.2522	0.4249	0.2879	0.2596
$b = 0.3$; $k_1 = 2$	0.2254	0.3556	0.2717	0.2440	0.2003	0.4200	0.2612	0.2430	0.3324	0.5625	0.3507	0.3851	0.2503	0.4325	0.2881	0.2582
$b = 0.75$; $k_1 = 1.2$ (default)	0.2152	0.3156	0.2514	0.2405	0.1791	0.3540	0.2381	0.2246	0.3336	0.5500	0.3558	0.3947	0.2436	0.4229	0.2838	0.2546

Proximity																
$p = 0.5$; $k_1 = 1.2$; $b = 0.1$	0.2394	0.3822	0.2775	0.2631	0.2088	0.4420	0.2683	0.2537	0.3319	0.5562	0.3481	0.3826	0.2703	0.4454	0.3060	0.2769
$p = 1$; $k_1 = 2$; $b = 0.1$	0.2375	0.3711	0.2871	0.2609	0.2150	0.4400	0.2725	0.2572	0.3352	0.5583	0.3506	0.3865	0.2647	0.4442	0.3003	0.2715
$p = 1$; $k_1 = 1.5$; $b = 0.4$	0.2325	0.3511	0.2642	0.2498	0.2104	0.4240	0.2699	0.2503	0.3453	0.5646	0.3638	0.3927	0.2709	0.4566	0.3067	0.2762
$p = 0.75$; $k_1 = 1$; $b = 0.3$	0.2332	0.3600	0.2782	0.2581	0.2083	0.4440	0.2722	0.2516	0.3387	0.5646	0.3593	0.3907	0.2751	0.4566	0.3112	0.2816
$p = 0.5$; $k_1 = 0.75$; $b = 0.1$	0.2362	0.3911	0.2769	0.2621	0.2002	0.4500	0.2616	0.2486	0.3233	0.5396	0.3395	0.3798	0.2701	0.4422	0.3020	0.2762
$p = 0.75$; $k_1 = 0.75$; $b = 0.2$	0.2311	0.3800	0.2795	0.2564	0.2012	0.4560	0.2633	0.2481	0.3303	0.5479	0.3480	0.3870	0.2733	0.4542	0.3083	0.2793
$p = 0.75$; $k_1 = 2.5$; $b = 0.4$	0.2292	0.3533	0.2652	0.2468	0.2083	0.4180	0.2660	0.2490	0.3430	0.5708	0.3635	0.3912	0.2607	0.4442	0.2977	0.2661
Lexical bonds																
$n = 1$; $k_1 = 2$; $b = 0.2$	0.2405	0.3667	0.2780	0.2587	0.2046	0.4340	0.2630	0.2445	0.3308	0.5771	0.3536	0.3859	0.2594	0.4450	0.2936	0.2649
$n = 0.5$; $k_1 = 1.5$; $b = 0.2$	0.2371	0.3622	0.2815	0.2578	0.2068	0.4380	0.2678	0.2461	0.3333	0.5729	0.3551	0.3858	0.2630	0.4442	0.2986	0.2690
$n = 0.25$; $k_1 = 1.2$; $b = 0.5$	0.2321	0.3489	0.2631	0.2524	0.1972	0.4060	0.2545	0.2374	0.3408	0.5667	0.3558	0.3919	0.2591	0.4394	0.2984	0.2664
$n = 0.5$; $\mathrm{k} = 1$; $b = 0.2$	0.2353	0.3622	0.2814	0.2587	0.2027	0.4500	0.2616	0.2449	0.3305	0.5792	0.3472	0.3826	0.2665	0.4514	0.3015	0.2733
$n = 0.25$; $k_1 = 1.2$; $b = 0.1$	0.2360	0.3711	0.2748	0.2603	0.2022	0.4320	0.2630	0.2444	0.3263	0.5562	0.3464	0.3815	0.2595	0.4434	0.2933	0.2664
$n = 1$; $k_1 = 1$; $b = 0.1$	0.2363	0.3689	0.2734	0.2596	0.2006	0.4580	0.2592	0.2448	0.3241	0.5604	0.3464	0.3806	0.2634	0.4450	0.2985	0.2698
$n = 0.25$; $k_1 = 1$; $b = 0.2$	0.2350	0.3667	0.2785	0.2588	0.2021	0.4420	0.2615	0.2434	0.3315	0.5792	0.3475	0.3829	0.2650	0.4486	0.3004	0.2721
Combined																
$n = 1$; $p = 0.75$; $k_1 = 1.5$; $b = 0.1$	0.2421	0.3822	0.2753	0.2629	0.2124	0.4680	0.2739	0.2552	0.3358	0.5646	0.3533	0.3865	0.2714	0.4570	0.3062	0.2781
$n = 0.5$; $p = 1$; $k_1 = 2$; $b = 0.1$	0.2392	0.3689	0.2860	0.2600	0.2154	0.4440	0.1355	0.2733	0.3370	0.5625	0.3524	0.3880	0.2677	0.4498	0.3039	0.2743
$n = 0.5$; $p = 1$; $k_1 = 2$; $b = 0.4$	0.2344	0.3444	0.2663	0.2518	0.2091	0.4240	0.2624	0.2475	0.3458	0.5708	0.3622	0.3925	0.2668	0.4458	0.3022	0.2720
$n = 0.5$; $p = 0.75$; $k_1 = 1$; $b = 0.2$	0.2350	0.3778	0.2767	0.2586	0.2075	0.4520	0.2721	0.2525	0.3360	0.5688	0.3542	0.3891	0.2753	0.4578	0.3097	0.2817
$n = 0.25$; $p = 0.5$; $k_1 = 1.2$; $b = 0.3$	0.2357	0.3556	0.2824	0.2587	0.2106	0.4380	0.2721	0.2532	0.3416	0.5667	0.3607	0.3917	0.2747	0.4635	0.3107	0.2811
$n = 0.5$; $p = 0.75$; $k_1 = 1.2$; $b = 0.1$	0.2401	0.3889	0.2827	0.2638	0.2101	0.4500	0.2707	0.2544	0.3328	0.5604	0.3486	0.3846	0.2712	0.4562	0.3081	0.2779
$n = 1$; $p = 1$; $k_1 = 1.5$; $b = 0.1$	0.2419	0.3800	0.2753	0.2627	0.2126	0.4700	0.2739	0.2554	0.3357	0.5667	0.3535	0.3863	0.2711	0.4574	0.3055	0.2779
$\mathrm{n} = 0.5$; $p = 1$; $k_1 = 1$; $b = 0.4$	0.2333	0.3511	0.2655	0.2533	0.2064	0.4320	0.2676	0.2502	0.3409	0.5771	0.3599	0.3923	0.2739	0.4534	0.3096	0.2803

References

Allan, J. (2005). HARD track overview in TREC 2004. High accuracy retrieval from documents. In E. Voorhees & L. Buckland (Eds.), *Proceedings of the 13th text retrieval conference* Gaithersburg, MD, USA.

Allan, J. (2006). HARD track overview in TREC 2005. High accuracy retrieval from documents. In E. Voorhees & L. Buckland (Eds.), *Proceedings of the 14th text retrieval conference* Gaithesburg, MD, USA.

Buckley, C., & Waltz, J. (2000). SMART in TREC 8. In E.M. Voorhees & D.K. Harman (Eds.), *Proceedings of the 8th text retrieval conference (TREC-8)* (pp. 577–582). Gaithersburg, MD, NIST, 2000.

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

Clarke, C. L. A., & Cormack, G.V. (1995). On the use of regular expressions for searching text. University of Waterloo Computer Science Department, Technical Report number CS-95-07, University of Waterloo, Canada.

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

Cormack, G. V., Clarke, C. L. A., Palmer, C. R., & Kisman, D. I. E. (2000). Fast automatic passage ranking (Multitext experiments for TREC-8). In E.M. Voorhees & D.K. Harman (Eds.) *Proceedings of the 8th text retrieval conference (TREC 1999)* (pp. 735–742). Gaithersburg, MD, USA.

Ellman, J., & Tait, J. (1998). Meta searching the web using exemplar texts: Initial results. In *Proceedings of the 20th BCS-IRSG conference*, 1998.

Fagan, J. L. (1989). The effectiveness of a nonsyntatic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science, 40*(2), 115–132.

Halliday, M. A. K., & Hasan, R. (1976). *Cohesion in English*. Longman.

Hawking, & Thistlewaite, P. (1996). Proximity operators – so near and yet so far. In D.K. Harman (Ed.), *Proceedings of the 4th text retrieval conference (TREC 1995)* (pp. 131–143). Gaithersburg, MD, USA.

Hoey, M. (1991). *Patterns of lexis in text*. Oxford University Press.

Hoey, M. (2005). Lexical priming. A new theory of words and language, Routledge.

Hull, D., Grefenstette, G., Schulze, M., Gaussier, E., Schütze, H., & Pedersen, J. (1997). Xerox TREC-5 site report: Routing, filtering, NLP and Spanish tracks. In E.M. Voorhees & D.K. Harman (Eds.), *Proceedings of the 5th text retrieval conference (TREC 1996)* (pp. 167–180). Gaithersburg, MD, USA.

Ishikawa, K., Satoh, K., & Okumura, A. (1998). Query term expansion based on paragraphs of the relevant documents. In E.M. Voorhees & D.K. Harman (Eds,), *Proceedings of the 6th text retrieval conference (TREC 1997)* (pp. 577–584). Gaithersburg, MD, NIST.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, Massachusetts: MIT Press.

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

Mitra, M., Buckley, C., Singhal, A., & Cardie, C. (1997). An analysis of statistical and syntactic phrases. In *Proceedings of RIAO-97* (pp. 200–214). Montreal, Canada.

Morris, J., & Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics, 17*, 21–48.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Rasolofo, Y., & Savoy, J. (2003). Term proximity scoring for keyword-based retrieval systems. In *Proceedings of the 25th European conference on information retrieval research. Lecture Notes in Computer Science* (vol. 2633, pp. 207–218). Pisa, Italy: Springer-Verlag. April.

Robertson, S. E., Zaragoza, H., & Taylor, M. (2004). Simple BM25 extension to multiple weighted fields. In *Proceedings of the 13th ACM CIKM conference* (pp. 42–49). Washington, DC, USA.

Salton, G. (1971). *SMART retrieval system; experiments in automatic document processing*. Englewood: Prentice-Hall.

Spärck Jones, K., Walker, S., & Robertson, S. (1998) A probabilistic model of information retrieval: Development and status, University of Cambridge Computer Laboratory Technical Report N 446.

Stairmand, M. A. (1997). Textual context analysis for information retrieval. In *Proceedings of the 20th ACM conference on research and development in information retrieval (ACM-SIGIR 1997)* (pp. 140–147).

Strzalkowski, T., Perez-Carballo, J., Karlgren, J., Hulth, A., Tapanainen, P., & Lahtinen, T. (2000). Natural language information retrieval: TREC-8 Report. In E.M. Voorhees & D.K. Harman (Eds.) *Proceedings of the 8th text retrieval conference (TREC 1999)* (pp. 381–390). Gaithersburg, MD, NIST, 2000.

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

Vechtomova, O. (2006). Noun phrases in interactive query expansion and document ranking. *Information Retrieval, 9*(4), 399–420.

Vechtomova, O., Karamuftuoglu, M., & Robertson, S. E. (2006). On document relevance and lexical cohesion between query terms. *Information Processing and Management, 42*(5), 1230–1247.

Voorhees, E. (2006). Overview of the TREC 2005 robust retrieval track. In E. Voorhees & L. Buckland (Eds.) *Proceedings of the 14th text retrieval conference* NIST, Gaithersburg, MD, November 2005.

Voorhees, E., & Buckland, L. (2004). In *Proceedings of the 12th text retrieval conference* NIST, Gaithersburg, MD, November 2003.

Xu, J., & Croft, B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th ACM conference on research and development in information retrieval (ACM-SIGIR 1996)* (pp. 4–11). New York: ACM Press.