Information Processing and Management 50 (2014) 568–583

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Automatic suggestion of phrasal-concept queries for literature search

Youngho Kim *, Jangwon Seo, W. Bruce Croft, David A. Smith

Center for Intelligent Information Retrieval, School of Computer Science, University of Massachusetts Amherst, 140 Governors Drive, Amherst, MA 01003, USA

ARTICLE INFO

Article history:
Received 20 September 2012
Received in revised form 15 March 2014
Accepted 17 March 2014

Keywords:
Query suggestion
Phrasal-concept query
Literature search

ABSTRACT

Both general and domain-specific search engines have adopted query suggestion techniques to help users formulate effective queries. In the specific domain of literature search (e.g., finding academic papers), the initial queries are usually based on a draft paper or abstract, rather than short lists of keywords. In this paper, we investigate phrasal-concept query suggestions for literature search. These suggestions explicitly specify important phrasal concepts related to an initial detailed query. The merits of phrasal-concept query suggestions for this domain are their readability and retrieval effectiveness: (1) phrasal concepts are natural for academic authors because of their frequent use of terminology and subject-specific phrases and (2) academic papers describe their key ideas via these subject-specific phrases, and thus phrasal concepts can be used effectively to find those papers. We propose a novel phrasal-concept query suggestion technique that generates queries by identifying key phrasal-concepts from pseudo-labeled documents and combines them with related phrases. Our proposed technique is evaluated in terms of both user preference and retrieval effectiveness. We conduct user experiments to verify a preference for our approach, in comparison to baseline query suggestion methods, and demonstrate the effectiveness of the technique with retrieval experiments.

1. Introduction

Query suggestion is an effective technique to help users by providing relevant query examples (Baeza-Yates, Hurtado, & Mendoza, 2004; Jones, Rey, Madani, & Greiner, 2006). This technique has been widely adopted for many Information Retrieval (IR) tasks, including domain-specific IR such as patent search and medical information retrieval (Kim, Seo, & Croft, 2011; Luo, Tang, Yang, & Wei, 2008). Literature search (e.g., finding relevant research papers) is one of the most promising domains that can be helped by query suggestion. In this domain, the typical users are scientists, and they need to find existing articles relevant to their current work. Since a scientific study is related to a number of research topics, people typically use many queries for retrieving a comprehensive list of related papers. In this situation, query suggestion can reduce the complexity of the search by providing effective query examples. In addition, sometimes scientists need to find relevant papers outside their specific area of expertise, and example queries can be a good guideline for exploring new areas.

Despite the potential effectiveness of query suggestion in literature search, there has not been much research in this area. Instead, prior work has focused on elaborating retrieval models that find related papers based on features derived from a detailed initial query, such as a draft paper (e.g., Bethard & Jurafsky, 2010; Larsen & Ingwersen, 2006; Ritchie, Robertson,

* Corresponding author. Tel.: +1 (413) 545 3059; fax: +1 (413) 545 1789.
E-mail addresses: yhkim@cs.umass.edu (Y. Kim), croft@cs.umass.edu (W.B. Croft).

http://dx.doi.org/10.1016/j.ipm.2014.03.003
0306-4573/

& Teufel, 2008; Strohman, Croft, & Jensen, 2007). Although these previous studies have found some important features for retrieval models (e.g., distance in a citation graph (Strohman, Metzler, Turtle, & Croft, 2005) and author citation behavior (Bethard & Jurafsky, 2010)), we believe that query suggestion can assist users in a complementary manner by making it easier to formulate multiple effective queries. Furthermore, we can integrate these queries with state-of-the-art retrieval models, leading to further improvement.

To develop effective suggestions for literature search, we need to consider its unique characteristics. In contrast to general web search, domain-specific search tasks (e.g., patent retrieval, medical search, and literature search) will each be carried out in a very specific environment, and a query suggestion method should be designed for the unique characteristics of that environment. In literature search, one unique characteristic is that phrasal concepts and terminology (e.g., "lexicon acquisition using bootstrapping") are frequently used as keywords in target documents (i.e., research papers). Since scientists use longer technical terms to describe their research ideas, phrasal concepts are frequently observed in academic writing. It follows that queries that emphasize phrasal concepts should be more effective for discriminating relevant documents from non-relevant documents in retrieval. In addition, typical users of literature search may prefer using phrasal-concept queries because phrases and terminology tend to have clear meanings, and users can more easily understand the areas that the suggested queries are targeting. Another typical characteristic of literature search is the lack of query log data. Many query suggestion methods for general web search rely heavily on large query logs (e.g., Baeza-Yates et al., 2004; Jones et al., 2006; Ma, Yang, King, & Lyu, 2008), but these data are not generally available for literature search because of a relatively small amount of search traffic.

Given that phrasal concepts are important for literature search and query log data is generally not available, we propose a query suggestion method that can generate phrasal-concept queries by exploiting pseudo-labeled documents. Specifically, we collect candidate (phrasal) concepts from pseudo-relevant documents (i.e., the top k documents retrieved by a baseline query), and identify "key concepts" – effective phrasal-concepts for finding relevant documents – by using the labeling propagation algorithm (Zhu & Ghahramani, 2002), which propagates retrieval effectiveness (labels) of the baseline query to associated candidate concepts. Once key concepts are found, we provide a context by extracting "related concepts" that are statistically associated with a key concept. We then construct phrasal-concept queries by combining key concepts with their related concepts, and suggest a list of phrasal-concept queries in descending order of predicted effectiveness of their key concepts. Note that we use the terms *phrasal-concept* and *concept* interchangeably.

For evaluation, the ideal situation is that scientists provide their research descriptions as initial queries, and relevant articles are identified by asking the same scientists. However, no such data is available, and there have been alternatives proposed to automatically generate evaluation data from existing citation databases (Ritchie, Teufel, & Robertson, 2006). For example, He, Pei, and Kifer (2010) develop an initial query using the sentences containing citations from a published paper, and regard the citations as the relevant articles. This approach favors a local recommendation because it only considers local contexts of the query paper (i.e., published paper) (He, Nie, Lu, & Zhao, 2012). On the other hand, the settings used in (Bethard & Jurafsky, 2010; Strohman et al., 2007) assume that the abstract and title of the query paper are a research summary written by the user, and the list of references cited in the paper is the set of relevant documents. This method uses the global context of the query paper for retrieval. In this paper, we adopt this approach for the retrieval experiments.

We evaluate our phrasal-concept query suggestion method based on user preference as well as retrieval effectiveness. We conduct user experiments to verify that users prefer the queries suggested by our technique, compared to other effective query suggestion and query expansion methods. To assess the retrieval effectiveness of our method, we compare the retrieval performance to other query expansion methods in simulated literature search environments.

The rest of this paper is organized as follows. In Section 2, we outline previous work in query suggestion, query expansion, and literature search. Section 3 defines a phrasal-concept query suggestion task for literature searches, and Section 4 describes our proposed techniques. In Section 5, we provide experimental results and discussions. We summarize our contributions and future work in Section 6.

2. Related work

2.1. Query suggestion

Query suggestion is a technique that recommends alternative queries for users' initial queries, which has been proven to be useful for improving users' search experience (Baeza-Yates et al., 2004; Jones et al., 2006). Since large scale web search engines can easily gather search logs that include user-issued queries and interaction information (e.g., clickthrough statistics), many previous suggestion techniques, especially for the web domain, use such resources (Baeza-Yates et al., 2004; Jiang & Sun, 2011; Jones et al., 2006; Ma et al., 2008; Mei, Zhou, & Church, 2008). For example Baeza-Yates et al. (2004) propose a clustering approach to extract similar queries from search logs for suggestion. They first cluster the aggregated queries using vector similarities, and identify the cluster that an initial query belongs to. Then, from the identified cluster, similar queries are extracted and suggested to the user. In a related approach Jiang & Sun, 2011) exploit query-hashing algorithms, which can map similar queries into the same hash code. Given search logs, they first generate prior-knowledge that indicates pairwise similarity and dissimilarity of the queries by using a hierarchical clustering, and formulate a hashing function which returns the same hash value for similar queries by minimizing the empirical error calculated using the prior-knowledge

(while minimizing the distance between similar queries and maximizing the gap between dissimilar queries). Jones et al. (2006) generate query suggestions by reformulating the original query using substitutions. They substitute initial query terms by synonyms, generalization, specification, and related terms. To do this, they develop a binary classifier which can predict the quality of a substitution, and train the classifier using manually labeled query pairs (i.e., <initial query, substituted (re-written) query>) extracted from query logs. Another approach for query suggestion uses clickthrough data to identify (semantically) related queries for an initial query (Ma et al., 2008; Mei et al., 2008). In this approach, two bipartite graphs (user-query and query-URL bipartite graphs) are formed using clickthrough data, and based on these bipartite graphs and a Markov random walk algorithm, a query similarity graph is generated and similarities between queries are propagated. Thus, given an initial query, similar queries, highly ranked by the similarity propagation of the initial query, are suggested to the user. In recent studies, the problem of diversifying suggestions is discussed (Ma, Lyu, & King, 2010; Song, Zhou, & He, 2011). In this work, alternative queries are extracted from query logs considering both relatedness to an initial query and diversification in the search results of the suggestions.

While query logs are readily available for web search, these data resources are generally not available in academic search environments. Some researchers have proposed query suggestion techniques that do not rely on query log data. For example Bhatia, Majumdar, and Mitra (2011) suggest relevant n-gram phrases for an initial query without using query logs. They extract n-grams from the corpus that are highly correlated with the partially input user query. In other words, relevant n-grams are suggested on the fly by completing the query that the user is typing. In our experiments, we use this approach as a baseline to compare with our approach. In the patent domain Kim et al. (2011) developed a Boolean query suggestion system that generates Boolean queries for an initial keyword query. They trained binary decision trees using the pseudo-relevant documents retrieved by the initial query, and extracted decision rules that determine whether a new document is pseudo-relevant or not. By doing this, a Boolean query can be formulated as a decision rule, i.e., a sequence of terms associated by conjunction where each term can be prefixed by negation.

2.2. Query expansion

Automatic query expansion (Mitra, Singhal, & Buckley, 1998) has been studied as a means to bridge the vocabulary gap between users' queries and relevant documents. In this process, initial queries are iteratively refined by including more terms that are potentially related to relevant documents. This area has been a focus of researchers for many years (e.g., Mitra et al., 1998; Xu & Croft, 1996). Since query suggestion also aims to find relevant queries (or terms) related to the initial query, query expansion is definitely related. However, the main difference is that query expansion methods place more emphasis on improving retrieval performance, while query suggestion techniques consider utility from the users' perspective as well as retrieval effectiveness.

Among many different approaches, Pseudo-Relevance Feedback (PRF) (Rocchio, 1971) is known as one of the most effective. This approach is based on the assumption that the top-ranked documents from an initial retrieval are relevant to the query. (Xu & Croft, 2000) extract expansion terms from the top retrieved documents for an initial query based on their co-occurrences with the initial query terms. Relevance models proposed by Lavrenko and Croft (2003) incorporate the pseudo-relevance assumption into the language modeling framework (Ponte & Croft, 1998). In this method, pseudo-relevant documents can be used for estimating a query model by deriving a multinomial distribution over the terms in pseudo-relevant documents. Terms that have high probability of occurrence in documents strongly related to the query are highly likely to be selected as expansion terms. The Latent Concept Expansion (LCE) method (Metzler & Croft, 2007) is the most closely related work to our method because it uses latent concepts extracted from pseudo-relevant documents to expand initial queries. However, this model works with short initial queries (e.g., the TREC query "hubble telescope achievements") whereas we assume that academic users provide longer queries which describe their new papers or projects. In addition, the Markov Random Field (MRF) framework used in (Metzler & Croft, 2007) was less effective using multi-term concepts (e.g., tri-grams) (Metzler & Croft, 2007) which frequently appear in academic articles. Another closely related study is the query expansion method proposed by Fonseca, Golgher, Possas, Riveriro-Neto, and Zibiani (2005). They view a past query in a query log as a concept, and past queries related to the current query are suggested to users to find more related concepts. However, their system relies on a sufficient volume of query log data, which cannot be easily acquired in typical small, domain-specific search systems. Other techniques related to our work involve modeling queries using concepts (e.g., Bendersky & Croft, 2008; Metzler & Croft, 2005). However, this research extracts concepts from only the initial query, and such concepts are less useful in our work because we focus on finding citations that can contain quite different terms from the initial query.

2.3. Literature search

Literature search is widely used by scientists for finding prior work that is relevant to their current research papers and projects. Previous studies in this area have improved retrieval models by extracting features from meta information (e.g., research interests of authors (Basu, Hirsh, Cohen, & Nevill-Manning, 2001)). For example Bradshaw, Scheinkman, and Hammond (2000) used only citing snippets to index articles cited at least once, and showed that this scheme could outperform a model that indexes the whole text of articles. In addition Ritchie et al. (2008) also showed that combining the citing snippets with the original documents can improve retrieval effectiveness, and (Larsen & Ingwersen, 2006) used citations of

the initial retrieval results in their experiments. Using the MEDLINE database,[1] (Meij & de Rijke, 2007) showed that weighting documents by the number of times they were cited can lead to gains in precision. In a different approach Strohman et al. (2007) used statistical learning frameworks to combine various meta features such as citation counts, common authors, and distances in a citation graph, and this approach achieved significant improvements over the baseline that used only simple keywords. More recently Bethard and Jurafsky (2010) developed citation behavior-based features. Since many academic authors appear to self-cite or cite the articles written by their co-authors, they proposed features that can boost the articles previously published by the query authors or their co-authors.

We improve literature search in several ways. First, we focus on making improvements to the query by providing query suggestions, whereas most existing work has elaborated retrieval models by meta information (i.e., information not based on the query text). Next, query suggestion techniques can be more practical and can help searches in real environments because the number of new articles is growing rapidly, and extracting all features from the corpus is complicated. Finally, the phrasal-concept queries generated by our method can be incorporated with existing retrieval features in the best models, which may lead further improvements.

Another piece of related work in literature search is context-aware citation recommendation. This task has been proposed by He et al. (2010), and assumes that an initial query of the query paper is the local context of a citation, which is the text surrounding the citation of the query paper. To solve this He et al. (2010) used a non-parametric probabilistic model which measures the similarity between a given context and target article by concept-based likelihood distribution. As another solution He et al. (2012) suggested a translation model to bridge the vocabulary gap between the context and retrieved documents. This type of local recommendation is effective if the authors describe detailed contexts for citations. In our work, we assume that the users provide a global context of the query paper by its title and abstract, and relevant articles (e.g., citations) are recommended by retrieval. Global recommendation is a different problem than local recommendation, and has been the focus of most prior work (e.g., (Bethard & Jurafsky, 2010)).

3. Problem formulation

In this section, we provide term definitions that we will use throughout this paper, and formulate the phrasal-concept query suggestion problem for literature search.

Definition 1 (*Literature search*). *Literature search* is a domain-specific search task that finds past published articles relevant to a new research work. This search task is helpful for scientists when they initiate new research projects or write up their work. In this paper, we assume that the users provide a research summary or paper abstract as an initial starting point, and focus on suggesting effective queries that can retrieve documents relevant to the research described in the summary.

Definition 2 (*Baseline query*). Given an initial query (e.g., a summary of a research work), a *baseline query* is its improvement by state-of-the-art query expansion methods (e.g., Latent Concept Expansion (Metzler & Croft, 2007)). We exploit the baseline query to generate more effective query suggestions.

Definition 3 (*Pseudo-relevant documents*). *Pseudo-relevant documents* are the top k documents retrieved by the *baseline* query. A state-of-the-art retrieval model is used to generate the ranking, and we extract phrasal concepts used for suggestions from the pseudo-relevant documents.

Definition 4 (*Phrasal concept*). A *phrasal concept* is a syntactic expression recognized as a noun phrase in a document. Syntactically-based phrases will be more recognizable to users in general than term sequences of some length (e.g., bigrams or trigrams). In addition, noun phrases are suitable for representing important "concepts" in academic papers (e.g., technique names such as "Markov Random Field"), and noun phrase concepts have been shown to be effective for improving retrieval effectiveness (Bendersky & Croft, 2008). In this paper, we use the terms *phrasal-concept* and *concept*, interchangeably.

Definition 5 (*Key concept and related concept*). A *key concept* is an effective phrasal-concept for finding relevant documents, and a *related concept* is a phrasal-concept related to a key concept, which helps users to understand the key concept better. For example, "text classification via WordNet" can be a key concept, and "Support Vector Machine" and "WordNet similarity feature" could be related concepts. A key concept can have multiple related concepts, and to measure the relation between a concept and the key concept, various statistical similarity measures can be used (see Section 4.2.2).

Problem 1 (*Key concept identification*). Given a set of phrasal concepts, *key concept identification* is ranking the concepts by their estimated retrieval effectiveness, i.e., highly ranked concepts are predicted to be more effective for retrieving relevant documents. We assume that the top n ranked concepts are the key concepts.

[1] PubMed (http://www.ncbi.nlm.nih.gov/pubmed/).

Definition 6 (*Phrasal-concept query*). A *phrasal-concept query* is a combination of a key concept and a set of related concepts. To improve the understandability of each suggestion and maximize retrieval performance, we include only a single key concept and its related concepts in a phrasal-concept query.

Problem 2 (*Phrasal-concept query suggestion*). *Phrasal-concept query suggestion* is suggesting a list of phrasal-concept queries to users. We suggest up to n queries which are sorted in descending order of predicted retrieval effectiveness of their key concepts. Since the key concepts in Problem 1 are ranked by their predicted retrieval effectiveness, we can address this problem by solving Problem 1.

4. Phrasal concept query suggestion

In this section, we describe our method to generate phrasal-concept queries. Given an initial query that the users input, we generate a list of n phrasal-concept queries in the following steps:

Step 1: Generate a baseline query, *BQ* and gather the pseudo-relevant documents of *BQ*.
Step 2: Extract candidate concepts from the pseudo-relevant documents.
Step 3: Identify n key concepts by ranking the candidate concepts using *BQ*. Related concepts may be also extracted.
Step 4: Construct a list of n concept queries as query suggestions.

The first step is improving the initial query to generate effective phrasal-concept queries. In this step, we use existing query expansion methods (e.g., Latent Concept Expansion (Metzler & Croft, 2007)) for the improvement. Since we assume that the users just input a bag of words as an initial query, such an initial query may perform poor and is not helpful for obtaining effective pseudo-relevant documents where phrasal concepts are extracted in the next step. To alleviate this, we use query expansion methods to generate a more effective set of pseudo-relevant documents. The query weighting schemes corresponding to the expansion method can also be applied. To formulate better baseline queries, we conducted preliminary experiments with several query expansion and generation methods and found that the LCE (Metzler & Croft, 2007) and machine learning-based approaches (Huang et al., 2006) performed significantly better in our search environments. So, in our experiments, we use these methods to generate baseline queries. However, any other query improvement method (e.g., relevance model (Lavrenko & Croft, 2003) or the dependence model (Metzler & Croft, 2005)) can be applied. Once a baseline query is formulated, we can obtain the top k pseudo-relevant documents from the retrieval result.

Next, we extract candidate (phrasal) concepts by ranking the phrases recognized from the pseudo-relevant documents. Then, in the third step, we rank the candidates with respect to their retrieval effectiveness predicted from the baseline query terms. After ranking, we assume that the top n (phrasal) concepts are key concepts, and combine each key concept with the related concepts that have high co-occurrence with the key concept. Finally, we can construct a list of phrasal-concept queries, each of which includes a single key concept and multiple related concepts. Fig. 1 shows an example of phrasal-concept query generation following this process, and the details of each step are described in the following sections.

4.1. Extracting candidate phrasal-concepts

In the second step, we collect candidate (phrasal) concepts used for identifying key concepts and their related concepts. By retrieving documents with the baseline query, we obtain pseudo-relevant documents, and then use them to extract candidate phrasal-concepts. As we consider a noun phrase (NP) as a phrasal concept (see Definition 4 in Section 3), we apply an NP recognizer[2] to the pseudo-relevant documents. However, due to the long length of academic articles (such as journal papers), too many phrasal-concepts are recognized from whole text of an article. Therefore, to reduce the size of the candidate set, we assume that a title and abstract contain important phrasal-concepts which can represent the whole article. Accordingly, we can generate two different candidate sets: (i) all phrasal-concepts from only the titles of pseudo-relevant documents and (ii) N important phrasal-concepts from titles and abstracts of pseudo-relevant documents; among all the recognized phrasal-concepts, we can use n-gram language models to estimate the importance of each phrasal-concept recognized from the titles and abstracts of pseudo-relevant documents. In the experiments, we use 300 phrasal-concepts extracted by using tri-gram language models. The ranking function based on the model is given as:

$$\begin{aligned} p(w_1 w_2 \dots w_l) &\approx \prod_{i=1}^{l} p(w_i | w_{i-1} w_{i-2}) \\ p(w_i | w_{i-1} w_{i-2}) &\approx \lambda_1 p(w_i | w_{i-1} w_{i-2}) + \lambda_2 p(w_{i-1} | w_{i-2}) + \lambda_3 p(w_{i-2}) \end{aligned} \tag{1}$$

where $w_1 w_2 \dots w_l$ is a concept whose word-length is l and λ_j is a bias to each language model.

[2] Montylingua (http://web.media.mit.edu/~hugo/montylingua/).

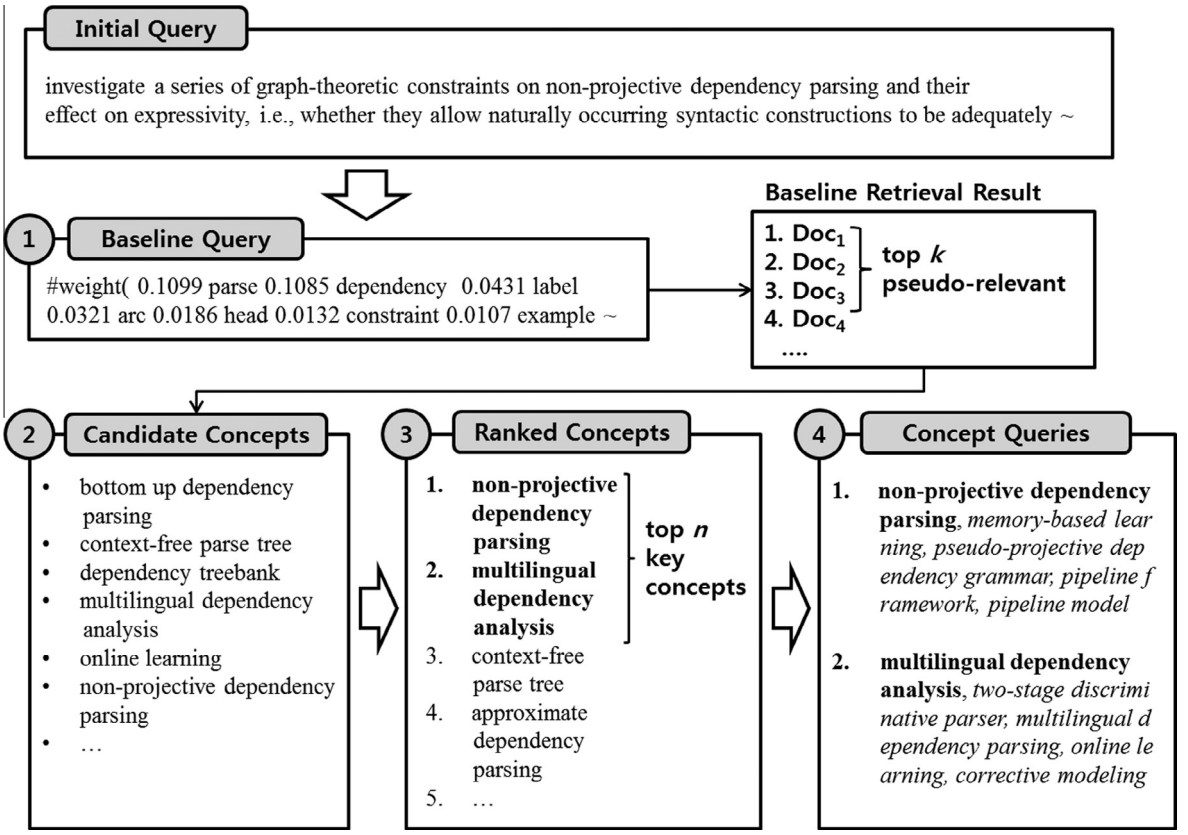

Fig. 1. Phrasal-concept query generation example. Bold in the step-3&4 indicates key concepts, and *italicized* in the step-4 denotes related concepts.

To avoid the sparseness problem, the tri-gram language models are smoothed by bigram and unigram language models, and for each model we use maximum likelihood estimations based on term frequencies in the pseudo-relevant documents. We empirically set the biases as $\lambda_1 = 0.7$, $\lambda_2 = 0.2$ and $\lambda_3 = 0.1$. If a phrasal-concept is longer than a tri-gram, we identify multiple tri-grams from the phrasal-concept (see the first part of Eq. (1)), and sum up the probability of each tri-gram to estimate the probability of the whole concept.

4.2. Identifying key phrasal-concepts

After collecting candidate phrasal-concepts, we identify key concepts by ranking the candidate (phrasal) concepts w.r.t. their predicted retrieval effectiveness. Given a set of candidate concepts and the baseline query, we assume that the concepts more similar to the baseline query will be more effective because the baseline query is effective for retrieving relevant documents. As an example, in Fig. 1, the initial query describes some graph-theoretic constraints for non-projective dependency parsing. In the baseline query, "dependency" and "parse" are effective keywords and highly weighted, and we can infer that among many related phrasal-concepts for this paper, "non-projective dependency parsing" is one of the most important phrasal-concepts. Since this phrasal-concept intuitively looks very similar to the keywords in the baseline query (i.e., "dependency" and "parse"), it may have higher retrieval effectiveness. To identify this phrasal-concept as a key concept, we use the similarity between the phrasal concept and keywords. Thus, in ranking, we place the phrasal concepts more similar to many baseline query terms at higher ranks, and the highly ranked phrasal concepts are regarded as "key concepts". To do this, we use the label propagation algorithm (Zhu & Ghahramani, 2002) where the labels (effectiveness) of the baseline query terms are propagated to the candidate concepts through a similarity matrix which defines the similarities between the candidate concepts and baseline query terms.

Suppose that we construct two vectors: (i) the vector of baseline query terms, $\boldsymbol{v_b}$ and (ii) the vector of candidate phrasal-concepts, $\boldsymbol{v_c}$. Define a term vector, $\boldsymbol{V}$, as $\boldsymbol{V} = [\boldsymbol{v_b}, \boldsymbol{v_c}]$ and construct a label vector $\boldsymbol{Y} = [\boldsymbol{y_b}, \boldsymbol{y_c}]$ where each $y_b \in \boldsymbol{y_b}$ is mapped to each $v_b \in \boldsymbol{v_b}$ and each $y_c \in \boldsymbol{y_c}$ is mapped to each $v_c \in \boldsymbol{v_c}$, i.e., $(v_1, y_1), (v_2, y_2), \ldots, (v_m, y_m)$ where $m = |\boldsymbol{V}| = |\boldsymbol{Y}|$. In addition, we define a $|\boldsymbol{V}| \times |\boldsymbol{V}|$ similarity matrix, $\mathbf{W}$ which represents the similarities between $\forall v_i$ and $\forall v_j$, i.e., $\mathbf{W}[i, j] = \text{sim}(v_i, v_j)$. To calculate $\text{sim}(v_i, v_j)$, we can use one of the following similarity measures.

Point-wise Mutual Information (PMI) is a statistical measure which quantifies the discrepancy between the co-occurrence probability in the joint distribution of v_i and v_j where the co-occurrence probability is estimated using their individual distributions. Using a corpus, the PMI of two terms (i.e., v_i and v_j) is calculated as:

$$\mathrm{PMI}(v_i, v_j) = \log \frac{p(v_i, v_j)}{p(v_i)p(v_j)} \approx \log \frac{\mathrm{df}(v_i, v_j) \times N}{\mathrm{df}(v_i)\mathrm{df}(v_j)} \tag{2}$$

where $v_i, v_j \in \boldsymbol{V}$, $\mathrm{df}(\cdot)$ denotes the document frequency in a corpus, and N is the number of all documents in the corpus.

Chi-square statistics (χ^2) is a statistical method that determines whether v_i and v_j are independent by comparing the observed co-occurrence frequencies with the expected frequencies assuming independence.

$$\chi^2(v_i, v_j) = \frac{(a \times d - b \times c)^2 \times N}{(a+b) \times (a+c) \times (b+d) \times (c+d)} \tag{3}$$

where $a = \mathrm{df}(v_i, v_j)$, $b = \mathrm{df}(v_i) - a$, $c = \mathrm{df}(v_j) - a$, and $d = N - a - b - c$.

Likelihood (LK) measures the likelihood of v_j to v_i, i.e., how much v_j can be generated from v_i. The calculation is given as:

$$\mathrm{LK}(v_i, v_j) = p(v_j|v_i) \approx \frac{\mathrm{df}(v_i, v_j)}{\mathrm{df}(v_i)} \tag{4}$$

Unlike the other measures, LK is directional, i.e., $\mathrm{LK}(v_i, v_j) \neq \mathrm{LK}(v_j, v_i)$.

With $\boldsymbol{V}$, $\boldsymbol{Y}$, and $\boldsymbol{W}$, we perform the concept ranking algorithm (Fig. 2) which produces a ranked list of the candidate (phrasal) concepts. In ranking, an initial output vector $\boldsymbol{Y}^{(0)}$ contains $\boldsymbol{y_b}$ corresponding to $\boldsymbol{v_b}$ and $\boldsymbol{y_c}$ corresponding to $\boldsymbol{v_c}$ where the values of $\boldsymbol{y_b}$ are 1.0 which indicates "labeled" (the highest retrieval effectiveness) and the values of $\boldsymbol{y_c}$ are 0 which indicates "unlabeled". Given a number of iterations (i.e., t), the propagation runs iteratively, and the values of y_c of phrasal concepts more similar to the baseline query terms may have higher values than the others less similar to the baseline query terms. Since t is a controlling parameter, if too high value of t is input, too many propagations are executed, and the values of $\forall v \in \boldsymbol{V}$ would be converged, i.e., the values of all candidate concepts are equal. Therefore, an appropriate value of t can be found by retrieval experiments (described in Section 5.1.2). After t iterations, the algorithm ranks $\boldsymbol{v}_c$ by the corresponding values of $\boldsymbol{y_c}$, and the phrasal concepts with greater values are placed at higher positions in the output list. In the output list, we assume that the top n phrasal concepts are "key concepts".

After identifying key concepts, we extract related concepts for each key concept. Since a similarity measure (e.g., PMI) can be defined between two phrasal concepts, we use it to extract "related concepts" among all candidate phrasal-concepts. In extraction, for each key concept, v_{KC}, we determine the set of "related concepts", $\boldsymbol{v_{RC}}$, as:

$$\boldsymbol{v_{RC}} = \{v | \mathrm{sim}(v_{KC}, v) > \theta\} \tag{5}$$

where θ is the cut-off value, v_{KC} is a key concept, v is a candidate phrasal-concept, $v_{KC} \neq v$. In the experiments, we empirically set θ as 0.01, 0.02, and 0.01 for PMI, χ^2, and LK, respectively.

Note that key concepts are identified as highly effective for retrieval, whereas related concepts are just strongly related to a key concept and provide additional context to the key concept for the users.

ALGORITHM Phrasal-Concept Ranking
INPUT:
- $\boldsymbol{V}$ is a input set divided into two sub-sets: the set of baseline query terms ($\boldsymbol{v_b} \in \boldsymbol{V}$) and the set of candidate phrasal-concepts ($\boldsymbol{v_c} \in \boldsymbol{V}$)
- $\boldsymbol{Y}$ is a label vector divided into two sub-sets: the set of baseline query terms ($\boldsymbol{v_b} \in \boldsymbol{V}$) and the set of candidate phrasal-concepts ($\boldsymbol{v_c} \in \boldsymbol{V}$)
- $\boldsymbol{W}$ is a similarity matrix which defines the similarities between $\forall v_i, \forall v_j \in \boldsymbol{V}$
- t is the number of iterations

OUTPUT:
- A ranked list of candidate concepts ($\boldsymbol{v_c}$)

PROCESS:
1. Let $\boldsymbol{D}$ be a diagonal and row sum matrix of $\boldsymbol{W}$
2. Initialize $\boldsymbol{Y}^{(0)} = [\boldsymbol{y_b}, \boldsymbol{y_c}]$ where $\boldsymbol{y_b} = \boldsymbol{1}$ and $\boldsymbol{y_c} = \boldsymbol{0}$.
3. For $i = 0, \ldots, t-1$
4. Calculate $\boldsymbol{Y}^{(i+1)} = \boldsymbol{D}^{-1} \cdot \boldsymbol{W} \cdot \boldsymbol{Y}^{(i)}$
5. End For
6. Sort $\boldsymbol{y_c}^{(t)} \in \boldsymbol{Y}^{(t)}$ in decreasing order
7. Return the list of $\boldsymbol{v_c}$ where the ranking of $v_c \in \boldsymbol{v_c}$ corresponds to the order of $y_c \in \boldsymbol{y_c}^{(t)}$

Fig. 2. Phrasal-concept ranking algorithm.

4.3. Constructing phrasal-concept queries

Given the top n key (phrasal) concepts, we construct n phrasal-concept queries by associating each key concept with its related concepts. As defined in Section 3, we ensure that a phrasal-concept query contains only a single key concept because a long query which contains several key concepts may be too complex to understand as a query suggestion. In addition, to further simplify the suggestions, we select the l most related concepts in the set of related concepts, $\boldsymbol{v}_{\boldsymbol{RC}}$ (see Eq. (5)). In the experiments, we empirically set l as 4, i.e., we make a query contain at most 5 phrasal-concepts including a key concept.

Finally, the n phrasal-concept queries are suggested to users, where each query is formed as $\langle$*Key Concept*, *Related Concept*$_1$, *Related Concept*$_2$, ...$\rangle$. The queries are listed in descending order of predicted retrieval effectiveness of their key concepts.

5. Experiments

This section describes evaluations for our method. In Section 5.1, we conduct retrieval experiments to verify the retrieval effectiveness of our approach, and Section 5.2 describes user experiments on preferences.

5.1. Retrieval experiments: literature search simulation

5.1.1. Experimental setup

In the experiments, we use MontyLingua[3] to identify phrasal concepts from the pseudo-relevant documents. Queries and documents are stemmed by the Krovetz stemmer. To simulate literature searches, we set up experimental environments as follows:

(Search Tasks) We conduct two different search tasks considering two domains of interest: the academic and medical domains. The task for the academic domain is finding academic papers relevant to a current research project. In this task, we assume that a scientist (user) inputs a summary of his research (i.e., we use title and abstract texts of his paper as the initial query) as an initial query, and we automatically generate a list of queries that can help to retrieve existing papers relevant to the research project. The search task for the medical domain is reference retrieval for an information need from physicians. We assume that physicians provide a statement of information about their patients as well as their information need, and we generate a list of queries that can retrieve relevant medical references for the information request.

(Collections) For our search tasks, we use two different collections consisting of academic and medical literature. For the academic literature collection, we used the ACL anthology corpus (Bird et al., 2008) which includes 10,921 academic papers published from 1975 to 2007. The full text of each article is available, and metadata (e.g., author names, venues, titles, and citations) is provided. We removed stop-words including frequently used acronyms (e.g., fig.) and section names (e.g., "introduction" and "related work") from the documents. To develop initial queries, we randomly selected 183 query papers published in 2006 from the collection, ensuring that their citations list contain at least 10 articles, and constructed an initial query by concatenating a query paper's title and abstract. As done in previous research (Bethard & Jurafsky, 2010; Ritchie et al., 2006; Strohman et al., 2007), we consider the articles cited in each query paper as "relevant" and 12.19 citations are listed on average. In addition, we discarded the references to articles outside of the collection that is searched, and the query papers are removed from the collection and relevance judgments for other papers.

For the medical literature collection, we used the OHSUMED collection (Hersh, Buckley, Leone, & Hickam, 1994), which consists of 348,566 medical references (abstracts) and 106 queries. Each query contains the statement of patient information and information need from physicians. This test collection contains relevance judgments manually annotated using three relevance levels (*definitely relevant*, *possibly relevant*, and *not relevant*). We consider *definitely* and *possibly relevant* as "relevant."

(Assumptions for Experiments) To implement a literature search simulation, we made the following assumptions. First, searchers directly use suggested queries without reformulation. We believe this helps to show the lower bound of performance that the proposed technique can achieve. Second, in a multiple-query session, searchers try the queries in the suggestion order. Since modeling user behavior is beyond the scope of this paper, we simply assume that searchers sequentially examine the queries starting from the first one.

(Evaluation Measures) To measure retrieval performance, we use traditional IR evaluation metrics as well as session-based measures. We adopt Mean Average Precision (MAP) and normalized Discounted Cumulative Gain (nDCG) (Jarvelin & Kekalainen, 2002) at the top 30 and 100 retrieval results. Also, normalized session Discounted Cumulative Gain (nsDCG) (Jarvelin, Price, Delcmbre, & Nielsen, 2008) is adopted for evaluating "session" retrieval results obtained by using multiple queries in a session. We use nsDCG to optimize our suggestion technique, and other traditional IR metrics (i.e., MAP and nDCG) for comparing our method with baselines. The metrics are calculated as follows.

First, MAP is defined in terms of Precision and Average Precision. Precision, P, is the fraction of retrieved results (documents) that are *relevant*, which can be calculated as:

[3] http://web.media.mit.edu/~hugo/montylingua/.

$$\mathrm{P}(R,D)=\frac{|R\cap D|}{|D|} \tag{6}$$

where D is the retrieved results and R is the set of relevant documents.

Average Precision, AveP, is the average of precision at each point where a relevant document is found and is computed as:

$$\mathrm{AveP}(R,D)=\frac{\sum_{i\in[1,|D|]:D_i\in R}\mathrm{P}(R,D_{[1:i]})}{|R|} \tag{7}$$

where D_i is an i-th ranked result in D.

Based on these, for a given set of queries, Q, MAP is calculated by:

$$\mathrm{MAP}(Q)=\frac{\sum_{q\in Q}\mathrm{AveP}(R_q,D_q)}{|Q|} \tag{8}$$

where q is a query in Q, R_q is the retrieval results of q, and D_q is the retrieved results of q.

Second, nDCG is defined by Discounted Cumulative Gain (DCG) which discounts the documents placed at the lower ranks in the retrieval list. The DCG of a particular rank, DCG@k, is defined as:

$$\mathrm{DCG@}k=rel_1+\sum_{i=2}^{k}\frac{rel_i}{\log_2(1+i)} \tag{9}$$

where rel_i is the relevance of the result at position i and $rel_i\in\{0,1\}$.

Using this, the nDCG at position k, nDCG@k, can be computed as:

$$\mathrm{nDCG@}k=\frac{\mathrm{DCG@}k}{\mathrm{IDCG@}k} \tag{10}$$

where IDCG is an ideal DCG score, i.e., when every relevant document is placed at the top of the retrieval list.

Third, we adopt a session-based metric that can measure the overall effectiveness of multiple queries because we suggest multiple queries for a search session. (Jarvelin et al., 2008) proposed the normalized session Discounted Cumulative Gain (nsDCG) which discounts documents that appear lower in a ranked list of an individual query as well as documents retrieved by the later suggested query. Given a session, nsDCG@k is calculated as follows.

First, a rank list is constructed by concatenating the top k documents from each ranked list of the session. For each rank i in the concatenated list, the discounted gain (DG) is computed as:

$$\mathrm{DG@}i=\frac{2^{rel_i}-1}{\log_2(1+i)} \tag{11}$$

where $rel_i\in\{0,1\}$.

We then apply an additional discount to documents retrieved by later suggestions. For example, the documents ranked between 1 and k are not discounted at all, but the documents ranked between $k+1$ and $2k$ are discounted by $1/\log_{bq}(2+(bq-1))$ where bq is the log base and determined by search behavior. A larger base, e.g., 10, indicates that a searcher is patient and willing to examine more suggestions, while a smaller base, e.g., 2, represents an impatient searcher. In this paper, we use $bq=10$ because academic searchers would use many queries to investigate more relevant articles. Then, Session Discounted Cumulative Gain (sDCG) at top k is calculated by:

$$\mathrm{sDCG@}k=\sum_{i=1}^{nk}\frac{1}{\log_{10}(j+9)}\mathrm{DG@}i \tag{12}$$

where $j=\lfloor(i-1)/k\rfloor$ and n is the number of suggestions (queries) in a session.

Accordingly, the final formula for nsDCG@k is given as:

$$\mathrm{nsDCG@}k=\frac{\mathrm{sDCG@}k}{\text{Ideal sDCG@}k} \tag{13}$$

where Ideal sDCG@k is an "ideal" score of sDCG obtained by an optimal ranked list in decreasing order of relevance.

(Retrieval Model) For retrieval, we implement a learning-to-rank retrieval model using SVM$^{\mathrm{rank}}$ (Joachims, 2006). This model can efficiently learn the weights of retrieval features from training data. To compose a feature vector in the retrieval model, we first use typical query-based features (e.g., *tf–idf* score) described in (Cao et al., 2006). In addition, we leverage meta features extracted from the document (e.g., age, venue, and citation information), proposed in (Bethard & Jurafsky, 2010) (see Table 1). The details of these features are described in (Bethard & Jurafsky, 2010). To improve the impact of concepts in phrasal-concept queries, we additionally created four concept-specific features (see Table 2). In total, 20 features are used for the learning-to-rank model.

(Baselines) Two different baseline approaches are employed for retrieval experiments. As baselines, we use the pseudo-relevance feedback techniques proposed in (Huang et al., 2006; Metzler & Croft, 2007), and the details of each method are summarized as follows.

Table 1
Retrieval features for learning-to-rank model. A t, q, d, and d_q indicate a term, query, target document, and the query paper where q is generated, respectively; freq(t, d) represents frequency of term t in document d; $idf(t)$ denotes inverse document frequency of term t; C denotes the entire collection; $|c|$ denotes the size of vocabulary in C. Query-based features are proposed in (Cao et al., 2006), and the details of meta features (i.e., Citation, Age, Citation Pattern, and Author Citation Behavior) are described in (Bethard & Jurafsky, 2010).

Category	Feature	Description
Query	$tf\,(q, d)$	$\sum_{t\in q\cap d}\log(\text{freq}(t,d)+1)$, frequency of query term
	$idf\,(q, d)$	$\sum_{t\in q\cap d}\log(\text{idf}(t))$, inverse document frequency
	tfidf (q, d)	$\sum_{t\in q\cap d}\log\left(\frac{\text{freq}(t,d)}{\lvert d\rvert}\text{idf}(t)+1\right)$, *tf–idf* score
	icf (q, d)	$\sum_{t\in q\cap d}\log\left(\frac{\lvert C\rvert}{\text{freq}(t,C)}+1\right)$, inverse collection term frequency
	lm (q, d)	$\sum_{t\in q\cap d}\log\left(\frac{\text{freq}(t,d)}{\lvert d\rvert}+1\right)$, unigram language model score
Citation	Citation-tfidf (q, d)	*tf–idf* score between q and all citations of d
Age	recency (d)	# of years since d was published
Citation Pattern	Citation-count (d)	# of times d was cited
	PageRank (d)	PageRank score of d in the citation network including all articles
	Venue-citation-count (d)	Citation count of articles published by the venue of d
Author Citation Behavior	Author-citation-count (d)	Citation count of the most cited author among authors of d
	Authors-self-cite (d_q, d)	Overlapping between authors of d_q and authors of d
	Authors-cited-article (d_q, d)	Overlap between authors of d_q and authors of articles citing d
	Authors–cited–author (d_q, d)	Overlap between authors of d_q and authors of articles citing articles by any author of d
	Authors–cited–venue (d_q, d)	Overlap between authors of d_q and authors of articles citing articles published by the venue of d
	Authors–coauthor (d_q, d)	Overlap between any authors of d_q and co-authors of d

Latent Concept Expansion (LCE) (Metzler & Croft, 2007) is a robust pseudo-relevance feedback technique based on a Markov Random Field framework. Comparing to relevance models (Lavrenko & Croft, 2003), this method is more generalized and can model term dependencies in a pseudo-relevance feedback process. To obtain feedback terms, we first obtain the top k pseudo-relevant documents (ranked using the sequential dependence model), and then the terms in the set of pseudo-relevant documents, R_D, are ranked by:

$$\text{LCE}(t)=\sum_{D\in R_D}\exp\left(\gamma_1 SD(Q,D)+\gamma_2\log\left((1-\alpha)\frac{tf(t,D)}{|D|}+\alpha\frac{tf(t,C)}{|C|}\right)-\gamma_3\log\frac{tf(t,C)}{|C|}\right) \tag{14}$$

where t is a feedback term, D is a document in R_d, Q is the initial query, $SD(Q, D)$ is a ranking score obtained by the sequential dependence model (Metzler & Croft, 2005), α is a smoothing parameter, $tf(t, D)$ is the term frequency in D, $tf(t, C)$ is the term frequency in a collection, C, and γ_i is a free parameter.

In this method, a feedback term is obtained by considering three features: (i) document relevance ($SD(q, d)$), (ii) term likelihood to the pseudo-relevant document model ($\log((1-\alpha)tf(t, D)/|D| + \alpha tf(t, C)/|C|)$), and (iii) dampening factor ($\log tf(t, C)/|C|$) to avoid highly common terms in C.

We select 10 documents for R_D and 80 (unigram) terms for feedback, and free parameters are set by 3-fold cross validation. In addition, we used bigrams for the feedback, but could not obtain any significant improvements relative to just using unigrams.

Table 2
Concept-specific features for learning-to-rank model. q indicates a query, d_q indicates the query paper where q is generated, and d indicates a target document.

Category	Feature	Description
Concept-specific	exist-key-concept (q, d)	Binary feature which returns 1 if d contains the key concept of q; otherwise, returns 0
	exist-all-concepts (q, d)	Binary feature which returns 1 if d contains all concepts of q; otherwise, returns 0
	loglike-key-concept (q, d)	Log-likelihood of q for d, estimated only by the key concept of q
		$\log\text{prob}(q\|d)\approx\log\frac{\text{freq}(kc,d)}{len_d-len_{kc}+1}$ where kc is a key concept of q, len_d is the length of d (# of words in d) and len_{kc} is the length of kc (# of words in kc)
	loglike-all-concepts (q, d)	Log-likelihood of q for d, estimated by every concept of q
		$\log\text{prob}(c\|d)\approx\sum_{c\in q}\log\frac{\text{freq}(c,d)}{len_d-len_c+1}$ where c is a concept in q, freq(c, d) is the frequency of c in d, len_d is the length of d (# of words in d) and len_c is the length of c

Machine Learning-based Expansion (MLE) is a method using a statistical learner for pseudo-relevance feedback, inspired by Huang et al. (2006) that exploits supervised learning algorithms. Given an initial query, to obtain a set of feedback terms, a linear regressor is trained with a set of features where each feature corresponds to a (unigram) term appearing in training documents (pseudo-relevant documents obtained by the initial query). Then, the trained regressor estimates the (pseudo-) relevance score of a new document, and the terms corresponding to highly weighted features are predicted to be effective for predicting pseudo-relevance. Note that this is a totally unsupervised procedure in that we do not use human-labeled samples.

We generate a set of training examples by using the top 100 pseudo-relevant documents and randomly sampled non-relevant documents which are not in the top 100 as positive and negative samples. We scale (pseudo) relevance to an interval [0, 1] and use them as target values in training. Specifically, we assume 11 different relevance degrees, i.e., {0.0, 0.1, 0.2, . . . , 1.0}, and generate 11 distinct sets, each of which contains an equal number of training examples where each set is mapped to the degree of the relevance; the top 100 pseudo-relevant documents are divided into the degrees from 0.1 to 1.0 (e.g., the top-1 to 10 documents are assigned to 1.0) and the beyond-100 documents are used for 0.0 (non-relevant). A feature set contains all words (except stop-words) from the pseudo-relevant documents, and a feature value is calculated by the *tf–idf* of a term in each document. After training, a weight vector, β is obtained, and among all components of β, we can select the top k features (terms) by ranking them in descending order of their weight values in β. To formulate an expanded query, the initial query is combined with the top k feedback terms, and the weight value from β is used for feedback term weighting. The bias to feedback terms against the initial query is set as 0.5, and 120 terms are selected as feedback terms. We tested this method with the features of noun phrases (longer than unigram), extracted from the training examples using a phrase recognizer.[4] However, the original setting of unigram terms could significantly outperform the case of noun phrases, and thus we use the unigram-based expansion as a more robust baseline.

5.1.2. Optimizing parameters

The first experiment is conducted to optimize the parameters of our method. In the phrasal-concept ranking algorithm (Fig. 2), the number of iterations and a similarity measure which defines a similarity matrix can influence the determination of key phrasal concepts. In addition, for academic literature search, we can use two different sets of candidates for ranking: (i) phrasal concepts only from titles of pseudo-relevant documents and (ii) phrasal concepts from titles or abstracts of pseudo-relevant documents (see Section 4.2.1). Thus, we test with different numbers of iterations, combinations of 2 candidate sets, and three different similarity measures. However, for medical reference retrieval, we use all phrasal concepts identified from pseudo-relevant documents because the OHSUMED collection does not provide section information, but the three different similarity measures can be tested.

Fig. 3 depicts the average nsDCG@100 over 1–20 iterations using the ACL collection. In retrieval, we used the Indri search engine (Strohman et al., 2005) to run the queries generated from each setting,[5] and 3-fold cross-validation was applied. For each session, we generated 10 phrasal-concept queries using the 6 different combinations. First, as the number of iterations increases, the performance reached a peak and afterward slightly decreases. Second, among the three proposed similarity measures, LK (likelihood) shows significantly better performance than PMI and χ^2. Third, the queries using the concepts from titles only (TTL) can reach the maximum more quickly and are slightly better than the queries using the concepts from titles or abstracts (TTL + ABST). This is because, in many papers, titles are sufficiently expressive while the abstract is often more verbose and noisy. To find an optimal combination, we compared the average nsDCG@100 of every combination, and the queries generated using TTL, LK and 5 iterations significantly outperformed most of the other cases (statistical significance in p-value < 0.05). Experiments using the OHSUMED collection showed similar tendencies.

5.1.3. Retrieval results

With the optimized parameters, we verify the retrieval effectiveness of our method on the two different search tasks. We use 3-fold cross-validation for evaluations, and LCE and MLE queries are used as baselines. As another baseline, we can consider the n-gram suggestion method (NGram) (Bhatia et al., 2011). However, we do not use it for this experiment because NGram focuses on finding relevant phrases for an initial query rather than improving their performance. Instead, we use that for user experiments (see Section 5.2). Besides, since the query expansion methods can significantly outperform n-gram suggestion in retrieval effectiveness, they can provide stronger baselines for retrieval experiments.

For academic literature search, we use the 20 features described in Tables 1 and 2 for our phrasal-concept queries, and the 16 features (Table 1) for baseline queries since the baseline queries do not contain phrasal concepts so we cannot use the 4 concept-specific features (Table 2). In the experiments of medical reference retrieval, we only use query-based features among the features in Table 1 because OHSUMED does not provide the meta information that is essential to implement non-query features (i.e., Age, Citation, Citation Pattern, and, Author Citation Behavior in Table 1). So, only 5 features (i.e., Query in Table 1) are used with LCE and MLE queries, and 4 concept-specific features are additionally included for phrasal-concept queries in OHSUMED experiments.

[4] Montylingua (http://web.media.mit.edu/~hugo/montylingua/).

[5] To run a phrasal-concept query, we use "#combine" for each phrasal-concept, as done in (Bendersky & Croft, 2009).

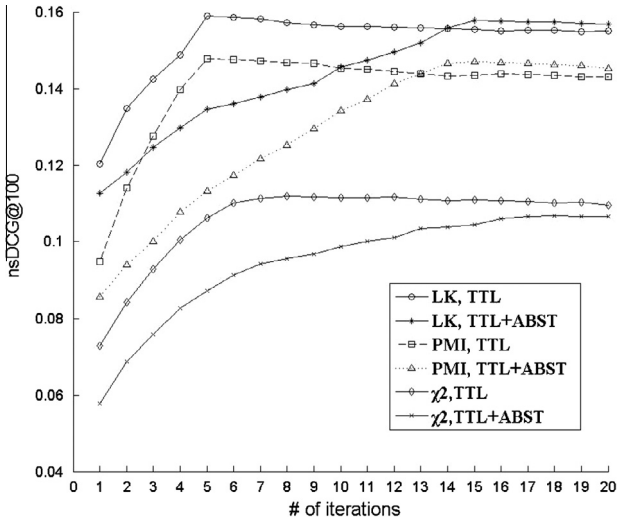

Fig. 3. nsDCG@100 of the top-10 concept queries using ACL collection. 'TTL' indicates concepts from the titles of pseudo-relevant documents. 'TTL + ABST' means concepts from the titles and abstracts of pseudo-relevant documents. LK, PMI, and χ^2 denotes the likelihood, PMI, and Chi-Square similarity measures, respectively.

To compare the performance between our method (PHRASAL-CONCEPT) and the baseline, we use the best average precision scores of the top-1 to 10 ranked phrasal-concept queries for each session, e.g., if the users browse the top 10 suggestions, we select the best query whose average precision score is the highest. Since our method generates multiple queries for a session, we select a single best query by the assumption that users examine the search results by all the top-*n* queries and identify the best query among them. In other words, we report an upper bound of the performance achieved by our method. Since authors sometimes need to use many queries to explore more relevant articles to their papers, browsing all of the top-*n* suggestions is not unusual, and they can subsequently recognize the most effective query among them. Besides, the baseline method can only generate a single best query, and the metric for multiple-query session (i.e., nsDCG) is not applicable.

Table 3 shows the average nDCG@100 and MAP of the results obtained by the best-performing query within the top-1 to 10 suggestions. First, in ACL, from the first suggestion, users can find an effective phrasal-concept query which can significantly outperform any baselines. Second, in OHSUMED, users need to examine the top two or more queries to find an effective phrasal-concept query that can perform significantly better than the best baseline (i.e., LCE). Third, phrasal-concept queries are significantly better than the baselines in most cases. Unlike the baseline queries, phrasal-concept queries can exploit the concept-specific features, and this leads to significant improvements over the baselines. For example, in Table 4, phrasal concepts in the concept query can effectively work with the concept-specific features for retrieval, whereas those

Table 3
Best query retrieval results for two different search tasks (ACL and OHSUMED). MLE and LCE indicate the baseline queries generated by Machine Learning-based Expansion (Huang et al., 2006; see Section 4.1) and Latent Concept Expansion (Metzler & Croft, 2007), respectively. Top-*n* denotes that among the top-*n* phrasal-concept queries, the best one is selected. In each column, a statistically significant improvement is marked using the first letter of each baseline method, e.g., ^{M} denotes a significant improvement over MLE. The paired *t*-test is performed with $p < 0.05$.

Collection	ACL		OHSUMED	
Method/Metric	nDCG@100	MAP	nDCG@100	MAP
LCE	0.4874	0.2638	0.4321	0.2748
MLE	0.5086	0.2744	0.4249	0.2660
PHRASAL CONCEPT				
Top-1	0.5301^{LM}	0.2899^{LM}	0.4328	0.2812^{M}
Top-2	0.5471^{LM}	0.3073^{LM}	0.4865^{LM}	0.3398^{LM}
Top-3	0.5626^{LM}	0.3211^{LM}	0.5236^{LM}	0.3737^{LM}
Top-4	0.5715^{LM}	0.3294^{LM}	0.5387^{LM}	0.3865^{LM}
Top-5	0.5780^{LM}	0.3364^{LM}	0.5505^{LM}	0.3973^{LM}
Top-6	0.5833^{LM}	0.3426^{LM}	0.5601^{LM}	0.4058^{LM}
Top-7	0.5873^{LM}	0.3473^{LM}	0.5643^{LM}	0.4097^{LM}
Top-8	0.5909^{LM}	0.3497^{LM}	0.5695^{LM}	0.4145^{LM}
Top-9	0.5933^{LM}	0.3518^{LM}	0.5748^{LM}	0.4183^{LM}
Top-10	0.5941^{LM}	0.3546^{LM}	0.5791^{LM}	0.4228^{LM}

Table 4
An example of an initial query, baseline query, and phrasal-concept query.

Initial query	Title: acquisition of verb entailment from text Abstract: the study addresses the problem of automatic acquisition of entailment relations between verbs. while this task has much in common with paraphrases acquisition which aims to discover ∼
Baseline query	Verb, emnlp, acquisition, entailment, semantic, pantel, related, text, deepak, value, special, grenoble, taxonomy, ∼
Phrasal-concept query	Paraphrases and textual entailment, generic paraphrase-based approach, semantic approach, relation extraction, entailment relation

features are not applied to the baseline query. This result is quite significant because we can identify that phrasal concepts can be new effective features for the literature search task, and are complementary to the previously developed features.

5.1.4. Further analysis

In Table 5, we show the number of improved or degraded queries w.r.t. the best baseline (i.e., MLE), within the top-10 suggestions for the 183 queries in the ACL collection. From this table, we can study the robustness of the proposed approach. About 70.6% of the queries generated by our method are more effective than the baseline. Moreover, about 44.4% of the generated queries dramatically outperform the baseline (i.e., improvements are greater than or equal to 25%).

5.2. User experiments: preference survey

In the user experiments, we conduct a questionnaire survey to identify preferences among a number of query suggestions. In other words, we ask users to select the most effective suggestion among many query examples generated by several methods. By doing this, we intend to identify which methods can generate more useful queries for users. We first describe the details of the survey, and then provide the results.

5.2.1. Survey settings

In our survey, we assume a situation where users (assessors) need to construct a list of articles relevant to a given paper (the "query" papers in our previous experiments). Each assessor is asked to select the most effective queries from the list of queries for finding the relevant articles. For each query paper, we first provide its title and abstract as a summary of the paper. Then, we list 8 different query suggestions generated by 4 different methods (NGram suggestion (NGram; Bhatia et al., 2011), Relevance Model (RM; Lavrenko & Croft, 2003), Machine Learning-based Expansion (MLE; Huang et al., 2006; see Section 5.1.1), and our method (PHRASAL-CONCEPT)) to an assessor. That is, two suggestions per method were provided. Finally, we ask them to select one or two queries that they believe would be more useful to retrieve relevant articles among the 8 suggestions. By doing this, the methods that can generate more effective queries for users would be chosen.

Fig. 4 shows an example of a question in the survey. To collect query papers, we selected 15 papers among the 183 query papers in our ACL collection (described in Section 5.1.1). For a fair comparison, the 15 papers were selected considering the results of retrieval experiments (Table 5); first, we selected 5 papers for which our proposed method worked significantly better than the baseline method in retrieval experiments (i.e., MLE); second, 5 papers were chosen for which the baseline method outperformed our method; finally, 5 papers were randomly selected among the papers for which our method performed as well as the baseline. This survey was done by the help of 20 volunteers who were graduate students majoring in computer science and familiar with the topics in computational linguistics (on which the ACL query papers focus). The details of each method are described as follows.

(NGram) While most existing query suggestion methods require query logs, the method proposed in (Bhatia et al., 2011) can suggest relevant n-grams without leveraging query logs. Since the original method aims at providing relevant n-grams when a user partially typed an initial query, we modify the method to fit in our search environments; we assume that a user finished typing the initial query and query completion is unnecessary. Note that this model is not used in the retrieval experiments (Section 5.1) because it focuses on suggesting correlated terms for an initial query rather than extracting effective ones for improving retrieval performance (as we explained in Section 5.1.3).

Similar to (Bhatia et al., 2011), given an n-gram, we use the log-likelihood ranking function based on phrase-query correlations.

$$\log p(Q_0|p_i) \approx \log \prod_{np \in Q_0} p(np|p_i) \approx \sum_{np \in Q_0} \log \frac{\mathrm{df}(np, p_i)}{\mathrm{df}(p_i)} \tag{15}$$

Table 5
# of improved or degraded concept queries w.r.t. Machine Learning-based Expansion (Huang et al., 2006; see Section 4.1) within top-10 suggestions using ACL collection. The number in parenthesis indicates the percentile ratio to the total number of queries.

Improved/degraded	$(\infty, -25\%]$	$(-25\%, 0\%)$	0%	(0%, +25%)	[+25%, ∞)	Sum
# of queries	139 (7.6%)	398 (21.8%)	0 (0.0%)	480 (26.2%)	813 (44.4%)	1,830 (100%)

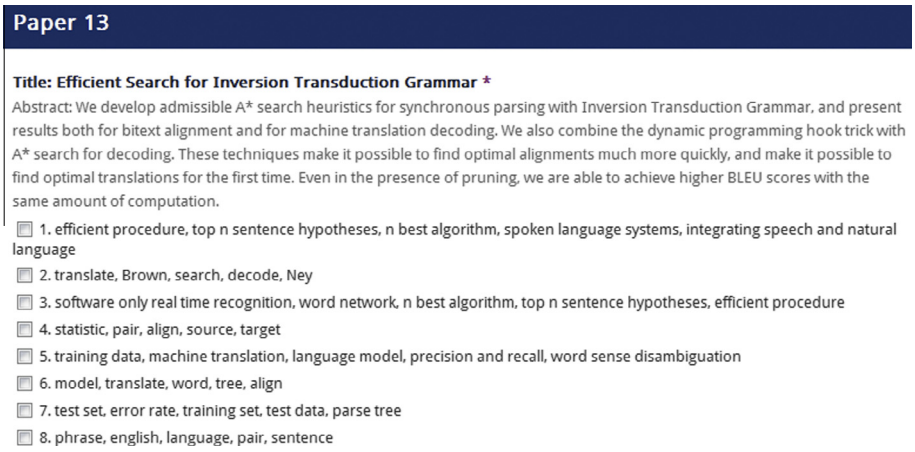

Fig. 4. An example question in user experiments.

where p_i is an n-gram phrase, *np* is a noun phrase and df$(\cdot)$ denotes the document frequency in a corpus.

For an initial query, Q_0, we use the title of a query paper, but in query ranking, as we see in Eq. (15), we count only noun phrases (longer than unigram) in Q_0 because counting correlation of every term in Q_0 is less efficient and noisy (e.g., the title texts contain less important terms such as "in" and "which"). We rank all n-grams of order 2, 3, 4, and 5 (i.e., bigrams to five-grams) from pseudo-relevant documents, and generate two queries by selecting the top-1 to 5 and top-6 to 10 n-grams ranked by this method.

The other baselines are query expansion methods proposed by Lavrenko and Croft (2003) and Huang et al. (2006), i.e., RM and MLE. For each baseline, we generate two different queries by selecting the top-1 to 5 and top-6 to 10 terms ranked by the method. We also use the top-1 and 2 phrasal-concept queries generated by our method with the optimal parameters (see Section 5.1.2). As a result, 8 queries are suggested, and to prevent assessors from inferring methods by the order of suggestions, we randomly shuffle the suggestion order.

5.2.2. Survey results and quality analysis

In the survey, a total of 484 responses was collected, and for each question (query paper), a respondent selected 1.61 queries on average, out of 8 queries (we asked to select only one or two of the best queries).

We first analyze the quality of queries generated by each method. Table 6 shows the top one and two suggested queries by each method for two research papers. First, it is clear that our phrasal-concept queries can present more plausible phrases than the baselines. For instance, "extracting structural paraphrases" refers to a task while "multiple sequence alignment" refers to a technique used in the field of paraphrase recognition (paper 1). Also, "extracting product features and opinions" and "learning subjective nouns" are important tasks in the study of opinion analysis (paper 2). Thus, these key concepts are related to many citations of each query paper. Second, the quality of NGram suggestions looks poor. Most of the suggested

Table 6
Examples of 8 suggestions generated by 4 different methods. The number in parenthesis indicates the number of responses which selected each method.

	Top-1 suggestion	Top-2 suggestion
Query Paper 1. Title: paraphrase recognition via dissimilarity significance classification		
RM (3)	Paraphrase, sentence, word, pair, translate	Phrase, match, align, extract, parallel
MLE (5)	Barzilay, paraphrase, align, synonymy, pair	Similar, regina, call, high, contiguous
NGram (0)	Noun phrase, artificial intelligence, training data, test set, machine translation	Machine learning, total number, statistical machine translation, human language technology
CONCEPT (31)	Extracting structural paraphrases, aligned monolingual corpora, paraphrase generation, large paraphrase corpora, multiple sequence alignment	Unsupervised construction, sentential paraphrases, exploiting massively parallel news sources, monolingual machine translation, paraphrase identification and corpus construction
Query Paper 2. Title: feature subsumption for opinion analysis		
RM (7)	Feature, word, sentence, set, opinion	Polarity, classify, term, train, data
MLE (0)	Feature, fix, Theresa, classify, classification	Set, class, recall, Joachim, manual
NGram (0)	Noun phrase, part of speech, training data, test set, machine learning	Supervised learning, error rate, statistical learning, number of words, set of features
CONCEPT (28)	Extracting product features and opinions, review classification via human provided information, extraction pattern bootstrapping, learning extraction patterns, learning subjective nouns	Phrase level sentiment analysis, contextual polarity, opinionated sentences, review classification via human provided information, subjectivity analysis

Table 7
Average number of responses for each method in a question. 20 assessors answered each question, and each assessor can choose one or two queries among 8 different suggested queries generated by 4 different methods. E.g., 20.87% of RM means that for a question, 20.87% of all 20 assessors prefer the query suggestions generated by RM. The statistically significance is marked using the first letter of each method (the paired t-test is performed with $p < 0.001$).

	RM	MLE	NGram	PHRASAL-CONCEPT	Sum
Response	6.74^{N}	4.93	0.73	19.87^{RMN}	32.27
Percentile ratio	20.87%	15.29%	2.27%	61.57%	100%

phrases are too general, and their meanings are vague since this method simply counts only correlations between the initial query and phrases without considering properties needed for queries in a specific domain. Another interesting point is that MLE tends to suggest the names of important authors who published frequently cited papers, e.g., "Regina Barzilay" (for paper 1) and "Theresa Wilson" (for paper 2). This is because MLE uses statistical leaning to extract highly discriminative terms, e.g., author name.

Next, we provide the average number of responses that selected queries generated by each method per question, as shown in Table 7. First, users strongly prefer to use our phrasal-concept queries, i.e., PHRASAL-CONCEPT accounted for 62% of the all responses. Second, although NGram can suggest phrases to the user, NGram suggestions are significantly less preferred because of their poor quality. As discussed above, the concepts suggested by our method look more readable and effective to retrieve relevant documents, and thus the assessors in the survey could show preferences on phrasal-concepts. However, user preferences in the survey may not reflect the exact effectiveness of suggestions in retrieval. Nevertheless, these preference results reveal that phrasal-concepts are more preferred by academic search users. Accordingly, our method is more useful than the baseline methods from the user perspective.

6. Conclusion

In this paper, we proposed a phrasal-concept based query suggestion technique for literature search. To generate more effective queries, we identified key concepts from pseudo-relevant documents by exploiting a label propagation technique and baseline query. By combining the key concept and its related concepts, a phrasal-concept query is generated. Through user studies and retrieval experiments, we showed that users strongly prefer to use our method and phrasal-concept queries can improve retrieval performance in literature search environments.

The merit of our approach is reproducibility and generalizability. To generate effective suggestions, we mainly use the concepts identified from pseudo-relevant documents, and similarities recognized within the corpus; any external resources or manually constructed data are not required. However, as Bai, Nie, Bouchard, and Cao (2007) studied, query contexts mined from external ontologies may help to identify more effective concepts and their relationships. Thus, for future work, we explore global information-based approaches applicable for the queries in academic literature search. In addition, we plan to use "semantic" concepts which cover semantic entities such as author names and domain-specific terminology in academic papers because such entities may be crucial to creating more effective and more "interesting" queries from the user's perspective.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Baeza-Yates, R., Hurtado, C., & Mendoza, M. (2004). Query recommendation using query logs in search engines. In *Proceedings of the 2004 international conference on current trends in database technology (EDBT'04)* (pp. 588–596).

Bai, J., Nie, J.-Y., Bouchard, H., Cao, G. (2007). Using query context in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'07)* (pp. 15–22).

Basu, C., Hirsh, H., Cohen, W. W., & Nevill-Manning, C. G. (2001). Technical paper recommendation: A study in combining multiple information sources. *Journal of Artificial Intelligence Research (JAIR), 14*, 231–252.

Bendersky, M., & Croft, W. B. (2008). Discovering key concepts in verbose queries. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'08)* (pp. 491–498).

Bendersky, M., & Croft, W. B. (2009). Analysis of long queries in a large scale search log. In *Proceedings of the 2009 workshop on web search click data (WSCD'09)* (pp. 8–14).

Bethard, S., & Jurafsky, D. (2010). Who should I cite? learning literature search models from citation behavior. In *Proceedings of the 19th ACM international conference on information and knowledge management (CIKM'10)* (pp. 609–618).

Bhatia, S., Majumdar, D., & Mitra, P. (2011). Query suggestions in the absence of query logs. In *Proceedings of the 34th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'11)* (pp. 795–804).

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., et al. (2008). The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of international conference on language resources and evaluation (LREC'08)*.

Bradshaw, S., Scheinkman, A., & Hammond, K. (2000). Guiding people to information: Providing an interface to a digital library using reference as a basis for indexing. In *Proceedings of the 5th international conference on intelligent user interfaces (IUI'00)* (pp. 37–43).

Cao, Y., Xu, J., Liu, T.-Y., Li, H., Huang, Y., & Hon, H.-W. (2006). Adapting ranking SVM to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'06)* (pp. 186–193).

Fonseca, B. M., Golgher, P., Possas, B., Riveriro-Neto, B. A., & Zibiani, N. (2005). Concept-based interactive query expansion. In *Proceedings of the 14th ACM international conference on information and knowledge management (CIKM'05)* (pp. 696–703).

He, Q., Pei, J., & Kifer, D. (2010). Context-aware citation recommendation. In *Proceedings of the 19th international conference on world wide web (WWW'10)* (pp. 421–430).

He, J., Nie, J.-Y., Lu, Y., & Zhao, W. X. (2012). Position-aligned translation model for citation recommendation. In *Proceedings of the 19th international conference on string processing and information retrieval (SPIRE'12)* (pp. 251–263).

Hersh, W. R., Buckley, C., Leone, T. J., & Hickam, D. H. (1994). OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'94)* (pp. 192–201).

Huang, X., Huang, Y. R., Wen, M., An, A., Liu, Y., & Poon, J. (2006). Applying data mining to pseudo-relevance feedback for high performance text retrieval. In *Proceedings of the sixth international conference on data mining (ICDM'06)* (pp. 295–306).

Jarvelin, K., Price, S. L., Delcmbre, L. M. L., & Nielsen, M. L. (2008). Discounted cumulated gain based evaluation of multiple-query IR sessions. In *Proceedings of the IR research, 30th European conference on advances in information retrieval (ECIR'08)* (pp. 4–15).

Jarvelin, K., & Kekalainen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS), 20*(4), 422–446.

Jiang, Q., & Sun, M. (2011). Fast query recommendation by search. In *Proceeding of the twenty-fifth AAAI conference on artificial intelligence (AAAI'11)* (pp. 1192–1197).

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'06)* (pp. 217–226).

Jones, R., Rey, B., Madani, O., & Greiner, W. (2006). Generating query substitutions. In *Proceedings of the 15th international conference on world wide web (WWW'06)* (pp. 387–396).

Kim, Y., Seo, J., & Croft, W. B. (2011). Automatic Boolean query suggestion for professional search. In *Proceedings of the 34th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'11)* (pp. 825–834).

Larsen, B., & Ingwersen, P. (2006). Using citations for ranking in digital libraries. In *Proceedings of the 6th ACM/IEEE-CS joint conference on digital libraries (JCDL'06)* (pp. 370–370).

Lavrenko, V., & Croft, W. B. (2003). Relevance models in information retrieval. In *Language modeling for information retrieval*. Springer.

Luo, G., Tang, C., Yang, H., & Wei, X. (2008). MedSearch: A specialized search engine for medical information retrieval. In *Proceedings of the 17th ACM international conference on information and knowledge management (CIKM'08)* (pp. 143–152).

Ma, H., Yang, H., King, I., & Lyu, M. R. (2008). Learning latent semantic relations from clickthrough data for query suggestion. In *Proceedings of the 17th ACM international conference on information and knowledge management (CIKM'08)* (pp. 709–718).

Ma, H., Lyu, M. R., & King, I. (2010). Diversifying query suggestion results. In *Proceeding of the twenty-fourth AAAI conference on artificial intelligence (AAAI'10)* (pp. 1399–1404).

Mei, Q., Zhou, D., & Church, K. (2008). Query suggestion using hitting time. In *Proceedings of the 17th ACM international conference on information and knowledge management (CIKM'08)* (pp. 469–478).

Meij, E., & de Rijke, M. (2007). Using prior information derived from citations in literature search. In *Proceedings of large scale semantic access to content* (RIAO'07) (pp. 665–670).

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 (SIGIR'05)* (pp. 472–479).

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 (SIGIR'07)* (pp. 311–318).

Mitra, M., Singhal, A., & Buckley, C. (1998). Improving automatic query expansion. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'98)* (pp. 206–214).

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'98)* (pp. 275–281).

Ritchie, A., Teufel, S., Robertson, S. (2006). Creating a test collection for citation-based IR experiments. In *Proceedings of the main conference on human language technology conference of the North American chapter of the association of computational linguistics (HLT-NAACL'06)* (pp. 391–398).

Ritchie, A., Robertson, S., & Teufel, S. (2008). Comparing citation contexts for information retrieval. In *Proceedings of the 17th ACM international conference on information and knowledge management (CIKM'08)* (pp. 213–222).

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In *The Smart retrieval system – Experiments in automatic document processing*. Prentice Hall.

Song, Y., Zhou, D., He, L.-W. (2011). Post-ranking query suggestion by diversifying search results. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'08)* (pp. 491–498).

Strohman, T., Metzler, D., Turtle, H., & Croft, W. B. (2005). Indri: A language-model based search engine for complex queries (extended version). Technical report, *UMASS CIIR*.

Strohman, T., Croft, W. B., & Jensen, D. (2007). Recommending citations for academic papers. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'07)*, pp. 705–706.

Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'96)* (pp. 4–11).

Xu, J., & Croft, W. B. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems (TOIS), 18*(1), 79–112.

Zhu, X., & Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical report, *CMU CALD*.