Learning to Refine Expansion Terms for Biomedical Information Retrieval Using Semantic Resources

Bo Xu, Hongfei Lin, and Yuan Lin

Abstract—With the rapid development of biomedicine, the number of biomedical articles has increased accordingly, which presents a great challenge for biologists trying to keep up with the latest research. Information retrieval seeks to meet this challenge by searching among a large number of articles based on given queries and providing the most relevant ones to fulfill information needs. As an effective information retrieval technique, query expansion has some room for improvement to achieve the desired performance when directly applied for biomedical information retrieval because there exist many domain-related terms both in users' queries and in related articles. To solve this problem, we propose a biomedical query expansion framework based on learning-to-rank methods, in which we refine candidate expansion terms by training term-ranking models to select the most relevant terms. To train the term-ranking models, we first propose a pseudo-relevance feedback method based on MeSH to select candidate expansion terms and then represent the candidate terms as feature vectors by defining both the corpus-based term features and the resource-based term features. Experimental results obtained for TREC genomics datasets show that our method can capture more relevant terms to expand the original query and effectively improve biomedical information retrieval performance.

Index Terms—Biomedical information retrieval, learning-to-rank, query expansion, term-ranking model

1 Introduction

In recent years, the rapid development of biomedicine has led to the treatment of intractable diseases and promoted the development of scientific research related to humans. At the same time, however, the explosive growth of information in biomedical articles poses a great challenge for biologists in obtaining all the articles related to one topic and thereby grasping the progress of research in their fields. Biomedical information retrieval is thus a hot research topic at the intersection of biomedicine and information retrieval (IR).

Given a query, biomedical information retrieval systems are designed to provide all relevant articles in a ranking list, in which articles are sorted based on their relevance to the query. The relevance can be determined using different retrieval models based on either the occurrences of query terms in the articles or probabilistic measures such as those used in language models. However, it is difficult to obtain optimal retrieval performance when directly applying these retrieval models to biomedical information retrieval. One possible reason for this problem lies in the incomplete interpretation of the information need of the user; that is, queries submitted by the user may partially express what he or she needs, thus resulting in failed retrieval. For example, given the biomedical query "How does P53 affect apoptosis?", the goal of the query is to find relevant documents focusing on the function of the protein P53 for apoptosis, and multiple aspects of the query should be covered in the search results such as apoptosis regulatory proteins, tumor suppressor protein P53 and gene expression, which can be used to enrich the query and better interpret the information need. Moreover, biomedical IR faces domain-specific challenges mainly caused by the abundant biomedical terms including synonyms and polysemy. Query expansion methods, as a series of classic and effective methods used in IR tasks [1], [2], [3], [4], can tackle the problem by enriching the original query with relevant terms and interpreting information needs to obtain more relevant articles.

Query expansion methods seek to reformulate the original query by adding relevant terms to better describe the information need, thus enhancing retrieval performance. These methods can be divided into two categories: corpus-based query expansion and resource-based query expansion. Corpus-based query expansion methods, such as the pseudo-relevance feedback (PRF) method, obtain expansion terms from the top-ranked documents in an initial retrieval with the assumption that the most frequent terms appearing in these documents are strongly correlated with the original query and can help improve the query. Resource-based query expansion methods employ external resources, such as domain-specific dictionaries, to measure the effectiveness of expansion terms. In biomedicine, there are many semantic resources, which contain many domain-specific terms and can be used to measure the importance of expansion

• *The authors are with the Dalian University of Technology, Room A923, Chuangxinyuan Building, No.2 Linggong Road, Dalian, Liaoning 116023, China. E-mail: xubo2011@mail.dlut.edu.cn, {hflin, zhlin}@dlut.edu.cn.*

Manuscript received 25 Oct. 2016; revised 23 Oct. 2017; accepted 26 Jan. 2018. Date of publication 2 Feb. 2018; date of current version 31 May 2019. (Corresponding author: Hongfei Lin.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2018.2801303

terms. Therefore, there is potential to improve the performance of biomedical IR by combining corpus-based query expansion and resource-based query expansion to find high-quality expansion terms.

Based on this idea, we propose a novel query expansion framework based on different learning-to-rank methods for biomedical information retrieval. Within this framework, we modify the pseudo-relevance feedback method to extract candidate expansion terms based on Medical Subject Headings (MeSH). To evaluate the usefulness of the candidate terms, we introduce learning-to-rank methods to perform supervised training of term-ranking models by defining both the corpus-based term features and the resource-based term features. The learned models are used to refine the expansion terms and assign weights to the selected expansion terms. Based on the selected terms, we reformulate the original query and retrieve with the expanded query to obtain better results. Experimental results obtained using TREC Genomics datasets show that our framework is effective in improving the performance of biomedical IR. We list the contributions of the paper as follows.

a) We combine MeSH-based term information with corpus-based term information to modify the pseudo-relevance feedback method and thereby find a large set of candidate expansion terms.
b) We define the corpus-based term features and the resource-based term features and introduce different learning-to-rank methods to refine the candidate terms by training term-ranking models.
c) We conduct extensive experiments to examine the effectiveness of our framework based on different learning-to-rank methods in comparison with different baseline models.

The remainder of the paper is organized as follows. In Section 2, we provide a review of related work. In Section 3, we introduce our query expansion framework in detail. In Section 4, we conduct extensive experiments to examine the effectiveness of the proposed method. In Section 5, we conclude the paper and provide suggestions for future work.

2 Related Work

In this section, we provide an overview of three areas of related work: research on query expansion, research on learning-to-rank, and research on query expansion for biomedical information retrieval.

2.1 Query Expansion

In information retrieval, query expansion is an effective and classic way to enrich users' queries with query-related terms, which can be integrated into different retrieval models, such as vector space [1], relevance [2], probabilistic [3], and mixture models [4]. Because the expansion terms help to better describe the information need, integrated models involving query expansion can improve retrieval performance.

Moreover, it has been observed that the quality of expansion terms can largely affect the effectiveness of query expansion. Therefore, some studies have attempted to refine the expansion terms using different methods. For example, Lee et al. [5] employed abundant linguistic and statistical term features to discover the underlying associations among expansion terms. Cao et al. [6] proposed classifying the candidate expansion terms to obtain high-query terms for expansion and demonstrated that informative term features can help choose good expansion terms in a supervised way. Based on these studies, ranking methods, particularly learning-to-rank methods, have been introduced for selecting high-quality expansion terms [7], which have demonstrated that ranking expansion terms according to their relevance to the original query can help refine the expansion terms and improve retrieval performance. In this paper, we attempt to modify the ranking-based query expansion method for biomedical information retrieval by taking domain-specific knowledge into consideration. Next, we briefly introduce learning-to-rank methods, which are the key technologies in our query expansion framework.

2.2 Learning-to-Rank

Learning-to-rank methods have been proposed and studied in recent years [8], [9], [10], [11], [12], [13]; these methods adopt supervised machine learning techniques to solve the ranking problem in IR tasks [14], [15]. Specifically, learning-to-rank methods modify the loss function of traditional supervised machine learning methods by incorporating ranking-based information and reduce the ranking loss iteratively in model training to construct the outputted ranking model. According to different forms of loss functions, learning-to-rank methods can be categorized into three approaches, namely, the pointwise approach, the pairwise approach and the listwise approach, which model the ranking loss for a single document, a pair of documents and a list of documents, respectively.

Because ranking is a central problem to be solved in many tasks, learning-to-rank methods have been introduced to achieve good performance in different tasks, such as community question answering [16] and recommendation system [17]. In our previous work [18], we proposed to optimize the pseudo-relevance feedback method, a classic query expansion method, using learning-to-rank methods to refine the set of expansion terms. In this paper, we further modify the framework to adapt it to biomedical information retrieval. Overall, there are two main differences between our previous work and this paper. First, we propose a novel MeSH-based pseudo-relevance feedback method by combining MeSH-based term information and co-occurrence-based term information. Second, we utilize domain-specific resources to extract effective term features for training the term-ranking models. In addition, we evaluate the modified query expansion framework on public available datasets through extensive experiments and demonstrate the effectiveness of the modified framework for biomedical information retrieval.

2.3 Query Expansion for Biomedical Information Retrieval

In biomedical information retrieval tasks, query expansion methods have been widely used to improve retrieval performance. As an initial attempt, Srinivasan [19] evaluated the retrieval effectiveness of query expansion strategies on a MEDLINE test collection using the SMART retrieval system. More recently, Xu et al. [20] compared local analysis, global analysis and ontology-based query expansion strategies for biomedical literature searches using TREC datasets. 'Matos et al. [21]

developed a new PubMed-based document retrieval and prioritization tool with a concept-oriented query expansion to find documents containing related concepts. Rivas et al. [22] studied query expansion techniques for biomedical information retrieval, including the retrieval of query-specific terms, corpus-specific terms and language-specific terms. These methods show that query expansion methods are useful for biomedical IR tasks and can be further enhanced with respect to certain domain-specific characteristics.

Furthermore, other studies have focused on latent concept expansion not only in the general IR field [23], [24] but also in the medical and clinical IR field [25], [26], [27], [28], [29]. In these studies, it has been proved that modeling latent query concepts has a significantly positive effect on retrieval performance. In the general IR field, Bendersky et al. [24] proposed assigning weights to concepts by applying the weighted dependence model to TREC corpora and web corpora, achieving better retrieval effectiveness. In TREC 2011&2012 medical records tracks, concept-based retrieval was also addressed to solve domain-specific tasks [25], [26]. In the clinical IR field, Zhu et al. [28] identified patient cohorts using mixtures of relevance models to weight query expansion terms. Wu et al. [29] proposed representing clinical queries as medically defined concepts for expansion and achieved modest improvements.

In addition, some studies have attempted to employ domain-specific resources, particularly the MeSH thesaurus, in query expansion processes to enhance retrieval performance. Drame et al. [30] proposed exploiting external resources to improve the performance of the vector space model in task 3 of the ShARe/CLEF eHealth Evaluation Lab 2014, where the MeSH thesaurus was used for query expansion with different configurations. Oh et al. [31] utilized external collections to optimize the PRF approach, incorporating the structure of external collections into their final feedback model. Mao et al. [32] integrated a MeSH-enhanced concept layer into a language modeling framework to make the most of concept associations. Jalali et al. [33] proposed a semantic query expansion method to match concept pairs between queries and the corresponding documents. Most studies have sought to optimize query expansion by capturing semantic information in biomedical resources directly. However, few studies have attempted to combine multiple term information for query expansion.

Learning-to-rank methods can integrate multiple term information by supervised learning and have been shown to be effective in many tasks. We have previously addressed biomedical information retrieval by modifying learning-to-rank methods for diversity-oriented passage retrieval [34]. In this paper, we explore the possibility of learning to rank a set of expansion terms for biomedical query expansion by extracting both corpus-based and resource-based term features.

3 Methods

In this section, we illustrate the general framework of our query expansion framework and provide detailed explanations of the proposed method, including the basic retrieval model, the method for choosing the candidate expansion terms, how the terms are represented as feature vectors for training the ranking models and the learning-to-rank methods for training the term-ranking models.

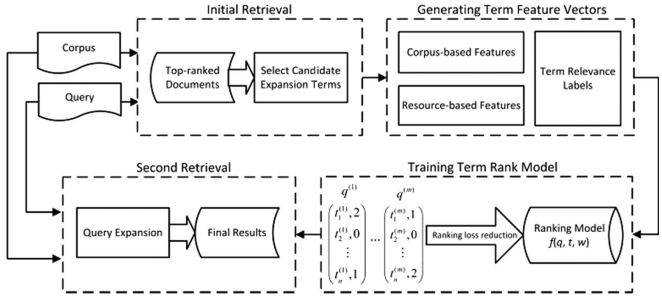


Fig. 1. Overall process flow of our query expansion framework.

3.1 General Framework

In this section, we introduce the general query expansion framework for biomedical information retrieval. Overall, the framework seeks to obtain a large set of candidate expansion terms from the top-ranked documents, namely, the feedback documents. We then utilize learning-to-rank methods to further refine the expansion terms and expand the query with the selected terms. Finally, we retrieve information with the expanded query to complete the retrieval process. We illustrate the process flow of our framework in Fig. 1.

The figure shows that we first conduct initial retrieval to obtain a large set of candidate expansion terms from the top-ranked documents, which are assumed to be relevant to the original query. Next, we represent these candidate terms as feature vectors including corpus-based features and resource-based features and assign each term a ground truth label according to its latent effect on retrieval performance. To train the term-ranking models, we conduct supervised training based on learning-to-rank methods with the term vectors and the ground truth labels as the training data. Finally, we use the model to refine the expansion terms for query expansion and retrieve with the expanded query.

3.2 Candidate Expansion Term Extraction

In this section, we introduce our method for choosing a large set of candidate expansion terms, which will be used later for term refinement of query expansion. Candidate expansion terms are extracted in two steps: First, initial retrieval is conducted to obtain feedback documents, which is mainly based on traditional language model of information retrieval; second, candidate terms are extracted based on the feedback documents, and we propose a novel method that considers both the feedback documents and Medical Subject Headings (MeSH) for measuring the relevance of terms.

3.2.1 Feedback Documents

To obtain the feedback documents, we conduct the initial retrieval with the original query based on the traditional language model of information retrieval using Dirichlet smoothing implemented in Indri [35], [36]. The language model is based on a probabilistic model for scoring documents with respect to one query and ranking the documents based on their relevance. After conducting the initial retrieval, we choose the top-ranked documents as the source of candidate expansion terms based on the pseudo-relevance feedback method. PRF assumes that these documents, namely, the feedback documents, are relevant to the original query, and the terms in these documents are closely

correlated to the original query, which can be used to enrich the query for interpreting the information needs.

3.2.2 Co-Occurrence-Based Candidate Term Selection

We first introduce a co-occurrence-based method [7] to select the candidate expansion terms from the feedback documents. Because high co-occurrence frequency indicates that two terms are strongly correlated, we take the co-occurrences of the candidate terms and the query terms as an indicator of term relevance. Namely, when one candidate term appears frequently with one query term in the feedback documents, we consider the term useful for enriching the query. We formulate this method as follows.

$$TFIDF_{DOC}(t,Q) = \sum_{q \in Q} idf_{DOC}(q) \cdot idf_{DOC}(t) \cdot \log\left(tf_{DOC}(t,q) + 1.0\right), \tag{1}$$

where t represents one of the terms in the feedback documents and q represents one of the query terms in a given query Q.$tf_{DOC}(t,q)$ is the co-occurrence term frequency of the term t and the term q, and $idf_{DOC}(t)$ is the inverse document frequency of the term t. These two items can be computed using the following equations.

$$tf_{DOC}(t,q) = \frac{\sum_{d \in D} \log\left(freq(t,d) + 1.0\right) \cdot \log\left(freq(q,d) + 1.0\right)}{\log|D|}, \tag{2}$$

where d represents one of the feedback documents D and $freq(t,d)$ is the raw term frequency of the term t in the document d. Similarly, $freq(q,d)$ is the raw term frequency of the query term q in the document d. $|D|$ is the total number of feedback documents.

Lin et al. [7] used social annotations to compute inverse document frequency. Because term occurrences in social annotations are sparser than those in the feedback documents we used, we adopt a widely used method [15] for computing idf in our method, which subtracts $n(t)$ from the numerator as follows.

$$idf_{DOC}(t) = \log \frac{N - n(t) + 1.0}{n(t) + 1.0}, \tag{3}$$

where N is the number of documents in the whole collection, and $n(t)$ is the number of documents containing the term t in the collection. $idf_{DOC}(t)$ measures the term importance in terms of its occurrences in the collection. To accumulate all the $tf_{DOC}(t,q)$ and $idf_{DOC}(t)$ with respect to the term t, we obtain the co-occurrence scores of t for the whole query in Eq. (1).

3.2.3 MeSH-Based Candidate Term Selection

In biomedicine, there are many semantic resources containing abundant domain-specific terms, which can potentially contribute to the query expansion process to choose more useful and relevant expansion terms. Controlled vocabularies, such as MeSH, have been proved effective in many biomedical information retrieval tasks. Therefore, we attempt to encode the MeSH-based information of terms into the co-occurrence method by considering term distribution in MeSH, which is similar to the method using term frequency and inverse document frequency. Specifically, we define the term frequency as the occurrences of terms in MeSH, and the inverse document frequency as the number of unique concepts containing the term in MeSH, because if a certain term appears in fewer concepts, the term possibly conveys more information than that appearing in more concepts. We formulate this approach as follows.

$$tf_{MeSH}(t) = \frac{\log\left(freq(t, MeSH) + 1.0\right)}{\log|T|}, \tag{4}$$

where |T| is the total number of terms in the MeSH thesaurus, and $freq(t, MeSH)$ is the raw frequency of the term t in MeSH.

$$idf_{MeSH}(t) = \frac{M - m(t) + 1.0}{m(t) + 1.0}, \tag{5}$$

where M is the total number of concepts in the MeSH thesaurus, and $m(t)$ is the number of unique concepts containing the term t.$idf_{MeSH}(t)$ can measure the importance of the term t in MeSH. Similarly to the computation in Eq. (3), we compute $idf_{MeSH}(t)$ based on the method defined in [15]. Finally, we combine $tf_{MeSH}(t)$ and $idf_{MeSH}(t)$ as follows:

$$TFIDF_{MeSH}(t) = idf_{MeSH}(t) \cdot \log\left(tf_{MeSH}(t) + 1.0\right). \tag{6}$$

Overall, we choose the candidate expansion terms via the co-occurrence-based method [7] and the MeSH thesaurus. We combine the two items from Eqs. (1) and (6) with a linear interpolation as follows:

$$score(t|Q) = \lambda \cdot \frac{TFIDF_{DOC}(t,Q)}{\sum_t TFIDF_{DOC}(t,Q)} + (1-\lambda) \cdot \frac{TFIDF_{MeSH}(t)}{\sum_t TFIDF_{MeSH}(t)}, \tag{7}$$

where λ is the interpolation parameter within the range of 0 and 1. We score all the terms in the feedback documents, rank the terms based on their scores from high to low, and choose the top-ranked terms as the candidate expansion terms for further refinement.

3.3 Term Feature Extraction

In this section, we represent candidate expansion terms as feature vectors. In a term feature vector, each dimension represents a term feature modeling the term in a unique manner, and the entire vector encodes the comprehensive information about the term, which can be useful for choosing the expansion terms. We divide the features into two categories: corpus-based features and resource-based features.

3.3.1 Corpus-Based Features

The corpus-based features are mainly based on the occurrences of terms in the corpus, which can be extracted based on different characteristics, such as term co-occurrence and term proximity. Moreover, the corpus-based features can be extracted from either the whole collection or the set of feedback documents. We provide detailed definitions of these features in the following sections.

Co-Occurrence-Based Term Features. The co-occurrence of terms refers to the frequency with which two terms appear in the same context, which can be an effective measurement of the similarity of two terms. Therefore, to measure the relevance between a given query and its corresponding expansion terms, we first accumulate the co-occurrences of a

query term and a candidate expansion term as a term feature as follows:

$$feature_1(t, Q) = \sum_{q \in Q} \sum_{d \in D} cooccurrence(q, t, d), \quad (8)$$

where Q is the original query and q is a query term in Q.D is the document collection and d is one document in D. In this feature, we count all the co-occurrences of term t with each query term q in all the documents in the entire corpus. Furthermore, we count the co-occurrences of one term with a pair of query terms as a stronger indicator of co-occurrence as follows:

$$feature_2(t, Q) = \sum_{(q_1, q_2) \in Q} \sum_{d \in D} cooccurrence(q_1, q_2, t, d), \quad (9)$$

where (q_1, q_2) can be any pair of query terms in the query.

Proximity-Based Term Features. Term proximity is a more focused measurement of term co-occurrence that counts the co-occurrence of two terms within a smaller distance, called the window size, instead of the whole document [37]. Therefore, proximity may be a more effective feature for measuring the relevance of candidate terms. We define term features based on proximity as follows:

$$feature_3(t, Q) = \sum_{q \in Q} \sum_{d \in D} \sum_{w \subseteq d} cooccurrence(q, t, w), \quad (10)$$

where w is the window size for measuring term proximities. We empirically set the window size from 1 to 10 words in our experiments.

Feedback Documents-Based Term Features. We also extract term features based on the feedback documents, namely, the top-ranked documents from the initial retrieval, because these documents are more likely to be relevant to the given query according to the basic assumption of pseudo-relevance feedback. These features may be of great use in characterizing the terms. Term features based on feedback documents can be categorized into two classes: features based on term frequency and inverse document frequency, which are classic statistics in information retrieval field, features based on co-occurrence and term proximity, where co-occurrences are measured in the set of feedback documents instead of the whole corpus.

3.3.2 Resource-Based Features

Because there are many semantic resources covering complex relationships of terms in the domain of biomedicine, we also seek to extract term features using these resources to model terms comprehensively, which may be useful for evaluating the importance of domain-specific candidate terms.

MetaMap-Based Features. We first attempt to extract term features based on the characteristics of biomedical concepts associated with the expanded query. Inspired by [38], we recognize the associated concepts using MetaMap [39], a biomedical natural language processing tool developed by the National Library of Medicine (NLM). MetaMap can discover concepts from a piece of biomedical text in the Unified Medical Language System (UMLS) metathesaurus. Specifically, we combine one candidate expansion term with the original query to form an expanded query and then convert the expanded query from a text query to a concept query, which contains the canonical forms of Concept Unique Identifiers (CUIs). Intuitively, if the concept query contains more biomedical concepts, it is more likely to convey useful information about the term, and the term may be more useful for expansion. We define three features based on this idea, as indicated in Eqs. (11), (12), and (13).

$$feature_4(t, Q) = count(t, Q'). \quad (11)$$

Eq. (11) counts the number of times that the term t appears in the concept query Q' of the expanded query with the term t and the original query Q.

$$feature_5(t, Q) = count_{CUI}(t, Q'). \quad (12)$$

Eq. (12) counts the total number of concepts for the concept query Q' of the expanded query, which can be a measurement of term importance at the query level.

Because the MetaMap program can return 1 to 10 candidates for each concept contained in a concept query, the number of returned candidates indicates the specific degree of the concept, which can reflect the importance of the term. We define a term feature based on this idea as follows:

$$feature_6(t, Q) = \frac{\sum_{c \in Q'} |R(c)|}{count_{CUI}(t, Q')}, \quad (13)$$

where $|R(c)|$ is the number of returned candidates for concept c of the concept query Q' of the expanded query with the original query Q and the term t. We normalize the feature values by the number of concepts contained in the concept query to make the feature values more comparable.

MeSH-Based Features. We also define term features based on their distribution in the MeSH thesaurus, including the number of occurrences of the term, the number of unique concepts containing the term, and the combination of these two factors. We have defined these features in the previous section, i.e., in Eqs. (5), (6) and (7) as $tf_{MeSH}(t)$, $idf_{MeSH}(t)$ and $TFIDF_{MeSH}(t)$, respectively. Because a given statistic will be more reliable after normalization, we normalize all feature values to the interval [0-1]. After representing the candidate expansion terms as feature vectors, we take these feature vectors as inputs, and the term labels as learning targets for learning-to-rank methods to construct term-ranking models in a supervised way.

3.4 Term-Labeling Strategy

After representing each candidate expansion term as a feature vector, we take the feature vectors as inputs for training the term-ranking model. Before training, we assign each term a ground truth label as the learning target. We take two factors into consideration when labeling a term: the term's impact on retrieval performance compared with the retrieval performance of the original query and the increasing magnitude of the performance. The labeling strategy is based on the following equation.

$$label_{term} = \begin{cases} 2 & eval(query + term) > eval(query) \text{ and } rank(term) \leq k \\ 1 & eval(query + term) > eval(query) \text{ or } rank(term) \leq k \\ 0 & eval(query + term) \leq eval(query), \end{cases} \quad (14)$$

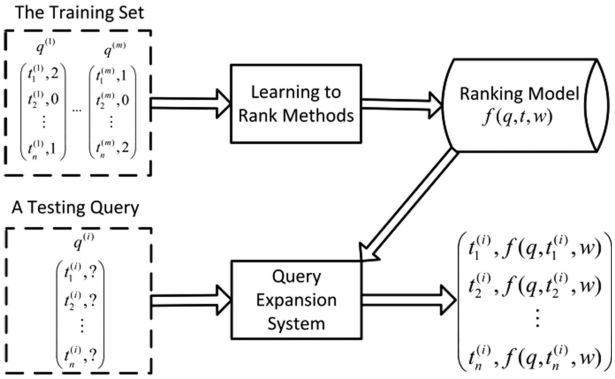


Fig. 2. Learning-to-rank framework for query expansion.

where *eval* can be any evaluation measures in information retrieval. Before assigning each candidate term a relevance label, we first retrieve using an expanded query with one term and the original query (denoted as *query+term* in Eq. (14)) and measure the difference in retrieval performance relative to the retrieval performance achieved with only the original query (denoted as *query* in Eq. (14)). We then sort the candidate terms based on their effects on the performance, and *rank*(*term*) indicates the rank of *term* in the sorted list. Namely, if the rank of a term is 1, the term contributes the most to the improved performance. Based on this idea, we label each term according to Eq. (14), where the label 2 means the term is definitely relevant to the original query, the label 1 means the term is possibly relevant to the query, and the label 0 means the term is irrelevant to the query. In our experiment, we choose the document mean average precision (Document MAP) as the *eval* function and tune the parameter k based on the retrieval performance of the original pseudo-relevance feedback method.

3.5 Learning-to-Rank Methods for Biomedical Query Expansion

To refine the expansion terms, we introduce learning-to-rank methods for term refinement by constructing term-ranking models. The training data for learning-to-rank are the defined feature vectors of the candidate expansion terms, with the ground truth labels of the candidate term as the learning targets. Learning-to-rank methods utilize machine learning techniques to solve ranking problems, which have been demonstrated to be effective in many IR tasks. The essence of learning-to-rank lies in incorporating ranking information into the loss function of supervised machine learning. The loss function for term ranking using learning-to-rank can be formalized as follows:

$$RankingLoss(Q,T) = \sum_{q \in Q} \sum_{i \subseteq t(q)} loss(i), \tag{15}$$

where Q is the query set, T is the overall candidate term set, and $t(q)$ is the candidate term set with respect to the query q. Eq. (15) indicates that the ranking loss is accumulated over all the queries in the training set.

For any query q, the ranking loss is counted by drawing different types of samples from the query and its corresponding terms. Samples can be categorized into three types: pointwise, pairwise and listwise, which take one document, a pair of documents and a list of documents into consideration when computing the ranking loss, respectively. A specific $loss(i)$ for a sample i is determined by the machine learning algorithms used. For example, for neural network-based learning-to-rank methods, $loss(i)$ is defined based on nonlinear combinations of different features. The final term-ranking model can be obtained by iteratively reducing the ranking loss using optimization methods, such as gradient descent.

To help understand the training process, we illustrate the learning-to-rank framework for query expansion in Fig. 2. The training set consists of queries, and each query corresponds to a set of candidate expansion terms with relevance labels. Learning-to-rank methods take the training set as inputs to learn the ranking model by iteratively reducing the ranking loss between the predicted relevance and the target relevance. In the testing phase, the learned model can be used to predict the relevance of the candidate terms for a testing query, ultimately choosing the expansion terms for query expansion.

In this paper, we use learning-to-rank methods for term refinement with the defined term features and use the constructed term-ranking models to choose the most relevant terms for the original query for expansion.

4 Experiments and Analysis

In this section, we conduct extensive experiments to examine the effectiveness of the proposed query expansion framework. We first introduce the experimental settings and the baseline models. Then, we evaluate the performance of the framework in terms of different feature sets and term-ranking accuracies. Next, we compare the retrieval performance of different ranking models, investigate the usefulness of term weighting, and compare the results of our method with the official results obtained for TREC Genomics tracks. Finally, we provide a comprehensive analysis of the experimental results and discuss our findings.

4.1 Experimental Setting

We examine our query expansion framework based on the datasets from TREC Genomics 2006 & 2007 tracks, which are publicly available datasets containing 162,259 articles from 49 genomics-related journals [40], [41]. The datasets contain 62 queries, among which 26 are from the 2006 track (we remove two queries with no relevant documents in advance) and 36 from the 2007 track.

We use the Indri search engine [36] as the basic retrieval system. We obtain N feedback documents from the initial retrieval and stem the terms in the documents with stopwords removed in advance. Based on the proposed method, we obtain candidate expansion terms for term refinement, k terms of which will be chosen in the final expanded query for second retrieval. The expanded query can be represented in the Indri query language as follows:

$$\begin{aligned} &\#weight(\alpha Q_{original}(1.0-\alpha)\#combine \\ &(\#weight(w_1 term_1 w_2 term_2 ... w_k term_k))). \end{aligned} \tag{16}$$

Eq. (16) corresponds to the expanded query with weighted expansion terms, where a is the weight on the original query, and the whole set of expansion terms is

TABLE 1
Retrieval Performance of Baseline Models on the 2006 Queries

Retrieval Model	Document	Passage	Aspect	Passage2
Language Model	0.3178†	0.0205	0.1983*	0.0239
Relevance Model	0.3194†	0.0207	0.2023*	0.0240
Term Dependency	0.3198†	0.0208	0.1785	0.0254
Cluster-based Model	0.3089	**0.0235***	**0.2644***	0.0258*
MeSH-based Model	0.3176†	0.0204	0.1902*	0.0241
Proposed PRF-weighted	0.3237*†	0.0212*	0.2037*	**0.0260***†
Proposed PRF-unweighted	**0.3242***†	0.0212*	0.2040*	**0.0260***†

'' indicates significant improvements over the term dependency method, and '†' indicates significant improvements over the cluster-based model.*

TABLE 2
Retrieval Performance of Baseline Models on the 2007 Queries

Retrieval Model	Document	Passage	Aspect	Passage2
Language Model	0.2587	0.0646	0.2000*	0.0876
Relevance Model	0.2678†	**0.0720***†	**0.2302***†	0.0963†
Term Dependency	0.2804†	0.0683†	0.1974	0.0939†
Cluster-based Model	0.2651	0.0673	0.1987*	0.0905
MeSH	0.2634	0.0706*†	0.2263*†	0.0941*†
Proposed PRF-weighted	0.2810*†	0.0705*	0.1995*	0.0991*†
Proposed PRF-unweighted	**0.2818***†	0.0706*	0.1996*	**0.0992***†

'' indicates significant improvements over the term dependency method, and '†' indicates significant improvements over the cluster-based model.*

weighted by (1.0- a). Each term in the expanded query is weighted using the ranking score obtained from the constructed term-ranking model. Because the ranking scores vary greatly among different ranking models, we normalize them to the range of 0 to 1.

$$\begin{aligned} &\#weight(\alpha Q_{original}(1.0-\alpha) \\ &\#combine(term_1 term_2 \ldots term_k)). \end{aligned} \tag{17}$$

Eq. (17) corresponds to the expanded query with unweighted expansion terms. We compare these two cases to examine the effect of term weighting on retrieval performance in the following section.

To obtain the average performance with learning-to-rank methods, we perform five-fold cross validations to train the term-ranking models. Specifically, we divide the queries from each dataset into a training set, a validation set and a test set in a ratio of 3:1:1, which follows the standard partition for the learning-to-rank datasets in LETOR [15]. We use the training set to train ranking models, the validation set to select the parameters for different ranking models, and the test set to predict new queries. We report the experimental results based on the average performance on all folds. We tune the parameters N, k, λ and a of the whole pseudo-relevance feedback for the 2007 queries with the 2006 queries and tune the parameters for the 2006 queries with the 2007 queries.

Because the TREC Genomics tracks design four evaluation measures to measure performance, we use these same measures in our experiments, namely, Document MAP, Aspect MAP, Passage MAP and Passage2 MAP. These measures modify the classic Mean Average Precision (MAP) to evaluate the retrieval performance from different perspectives for the biomedical information retrieval tasks [40], [41].

In the following sections, we first examine the effectiveness of the proposed method for candidate term selection, and then we train learning-to-rank based term-ranking models. Thereafter, we examine the performance of the term-ranking models in terms of ranking accuracy and retrieval performance. We also compare the term-ranking models based on the effect of term weighting on retrieval performance, and provide in-depth analysis and discussions.

4.2 Retrieval Performance of Baseline Models

In this section, we seek to examine the performance of the proposed pseudo-relevance feedback method for candidate term selection, which combines term information in the feedback documents and the MeSH thesaurus. The retrieval performance for the 2006 and 2007 queries are presented in Tables 1 and 2, respectively, compared with the performance measured for the baseline models.

The models compared in the tables include the query-likelihood language model (QL) [4] implemented in the Indri search engine and Lavrenko's relevance model (RM) [2], which expands the queries with the top-k most relevant terms obtained from N feedback documents. The third is the term dependency model [7], which is an effective modified pseudo-relevance feedback model that selects expansion terms by considering both the full independence and sequential dependence of the expansion term and the original query. The fourth is the cluster-based model [31], which uses k-means to choose expansion terms for medical information retrieval. For the proposed methods, the MeSH-based model refers to the proposed pseudo-relevance feedback model when scoring terms based solely on term distribution in MeSH. We compare the retrieval performances of our PRF methods in two situations, namely, when weighting the expansion terms with the PRF-based scores and when the terms shown in the tables are unweighted. We conduct two-tailed paired Student t-tests ($p < 0.05$) to examine whether the improvements are significant relative to the baseline models, where an asterisk indicates significant improvements over the term dependency method and a dagger indicates significant improvements over the cluster-based model.

Table 1 shows that the baseline models, including the relevance model and the term dependency model, outperform the basic retrieval with small improvements, and the retrieval model based solely on the MeSH-based PRF achieves a performance level comparable to that of these models. Furthermore, the proposed PRF method with unweighted expansion terms achieves the best retrieval performance in terms of Document MAP and Passage2 MAP, and the cluster-based model achieves the best performance in terms of the other two evaluation measures.

The results for 2007 queries in Table 2 show the same trend. As shown in the table, all of the baseline models yield much better results compared with those yielded by the basic language model. The relevance model achieves the best performance in terms of Passage MAP and Aspect MAP, and the proposed PRF methods significantly outperform the other methods with the weighted or the unweighted expansion terms in terms of Document MAP and Passage2 MAP.

The experimental results for both years' queries indicate that the proposed PRF method can improve retrieval performance, which would be highly useful for further term refinement. One possible explanation for this finding is that the

TABLE 3
Term-Ranking Accuracies with All the Defined Features on Both Query Sets

Ranking methods	For the 2006 queries	For the 2007 queries
SVM	0.9358	0.7489
MART	0.9342	0.7852
RankNet	0.8320	0.7119
RankBoost	0.9383*	**0.7868***
ListNet	0.7002	0.6539
LambdaMART	**0.9462***	0.7721*

'' indicates significant improvements over the SVM-based model.*

MeSH-based term information can enhance the co-occurrence-based retrieval model by incorporating the term distributions in the MeSH thesaurus, which helps measure the term importance for query expansion, particularly for choosing candidate domain-specific terms. Furthermore, we find that the term weights obtained from the proposed method are of little use for improving the performance, and even hurt the performance, which inspires us to develop a more effective method for term weighting for expanded queries.

4.3 Evaluations on Term-Ranking Accuracy

In our query expansion framework, we utilize the proposed MeSH-based PRF method to choose a large set of candidate expansion terms and represent each term as a feature vector based on the labeling strategy and the pre-defined term features. We take the feature vectors of terms in the training set as the inputs for learning-to-rank methods to train the term-ranking models; the models can thus be used to predict the term ranking for the queries in the testing set.

To choose more useful expansion terms and balance the importance of the expansion terms using term weighting, we introduce learning-to-rank methods to construct term-ranking models for term refinement. In our experiments, we examine six learning-to-rank methods for our query expansion framework, which adopt one of three approaches: the pointwise approach, the pairwise approach or the listwise approach. Different approaches to learning-to-rank adopt different sampling strategies to train the term-ranking models. The pointwise methods MART [8] and SVM [6] are designed to predict the relevance of each term; the pairwise methods RankNet [9] and RankBoost [11] seek to predict the preferred order of two terms with different relevance labels; and the listwise methods ListNet [10] and LambdaMART [12] take the term-ranking list as a whole unit to optimize the ranking list.

SVM-based term ranking [6] was first proposed to implement expansion term refinement to classify terms as good terms or bad terms and weight the terms based on their posterior probability. By contrast, we adopt learning-to-rank methods to refine the candidate expansion terms and weight the terms based on the ranking-based scores. We use this method as a strong baseline in our experiments.

In this section, we first evaluate the term-ranking accuracies of all the methods to examine their effectiveness for term ranking. Intuitively, the term-ranking models that exhibit the highest ranking accuracy can choose the most effective terms for query expansion and improve retrieval performance. Table 3 shows the term-ranking accuracies obtained from the ranking models constructed with all the defined term features on both query sets. We conduct two-tailed paired Student t-tests ($p<0.05$) to examine whether the improvements are significant relative to the results of the SVM-based ranking model, where an asterisk indicates that the improvements are statistically significant. The results have been averaged over all folds from cross validations.

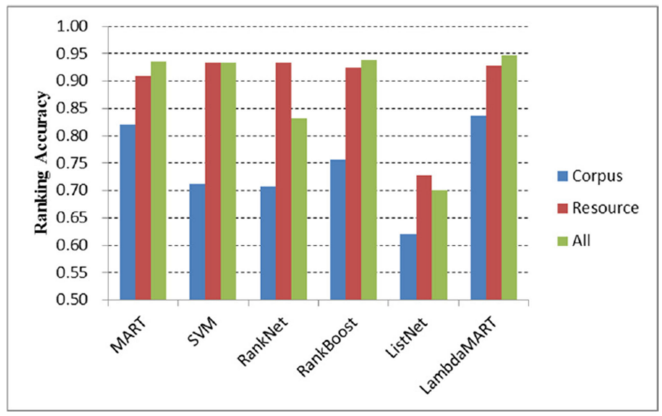


Fig. 3. Term-ranking accuracies on the 2006 queries.

The table shows that, on the 2006 queries, the ranking models based on SVM, MART and RankBoost achieve comparable results, outperforming the ranking models based on RankNet and ListNet, and the LambdaMART-based ranking model achieves the highest ranking accuracy. Similar trends can be observed for the 2007 queries, except that the RankBoost-based ranking model achieves the best performance. Because we define the term features based on corpora and other resources, we further examine the effectiveness of these two features sets for constructing the ranking models. Figs. 3 and 4 show the term-ranking accuracies with different feature sets for each learning-to-rank method on the 2006 queries and the 2007 queries, where "Corpus" refers to the ranking models trained with only corpus-based features (defined in Section 3-D-1)), "Resource" refers to the ranking models trained with only the resource-based features (defined in Section 3-D-2)), and "All" refers to the ranking models trained with all the defined features.

The figures show similar trends for both query sets. Ranking models based on the resource-based features yield better results than those based on the corpus-based features for

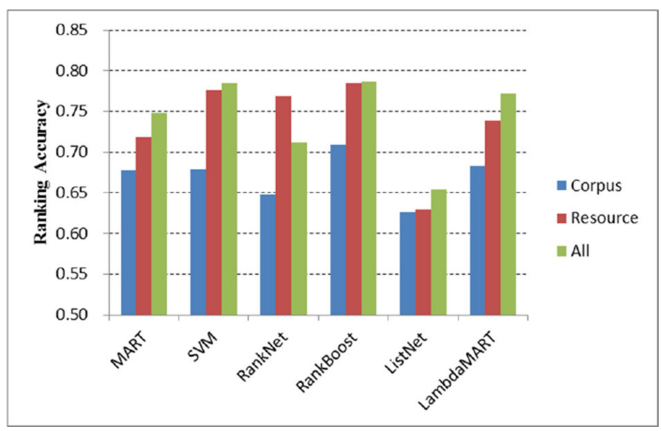


Fig. 4. Term-ranking accuracies on the 2007 queries.

TABLE 4
An Example on Top-10 Ranked Terms of Different Methods

SVM	Rel.	RankBoost	Rel.	LambdaMART	Rel.
disease	2	disease	2	disease	2
prions	1	prions	1	cow	1
cause	0	mad	1	encephalopathies	2
infectious	2	encephalopathies	2	infectious	2
conversion	0	neurodegenerative	2	spongiform	1
cow	1	cow	1	scrapie	1
spongiform	1	spongiform	1	prpc	1
fatal	0	virus	0	nucleic	1
encephalopathies	2	cause	0	cerebral	1
mad	1	prpc	1	virus	0

different learning-to-rank methods, and ranking models based on all the features outperform ranking models obtained using either the corpus-based features or the resource-based features.

To help understand the results of the term ranking, we provide an example of term ranking based on the query "What is the role of PrnP in mad cow disease?" (No.160 of 2006 queries). We list the top 10 ranked expansion terms yielded by the three methods in Table 4, where "Rel." represents the relevance label of each term.

The table shows that the LambdaMART-based model can output more accurate term rankings with definitely relevant terms at the top of the ranking list compared with the other two models while reducing the number of irrelevant terms in the top-ranked list. The phenomenon is consistent with the results in Table 3, based on which we can choose more relevant terms in the expanded query to enhance retrieval performance.

4.4 Comparisons of Retrieval Performance

After examining the term-ranking accuracy of the trained ranking models, we examine the effectiveness of the term-ranking model for refining the candidate expansion terms in the second retrieval. We compare the retrieval performance of query expansion using different term-ranking models in this section relative to the proposed PRF method. Tables 5 and 6 show the experimental results for both of the query sets. We conduct two-tailed paired Student t-tests ($p < 0.05$) to examine whether the improvements are significant relative to the baseline models, where an asterisk indicates significant improvements over the proposed PRF method and a dagger indicates significant improvements over the SVM-based model.

TABLE 5
Retrieval Performance with All the Features on the 2006 Queries

Methods	Document	Passage	Aspect	Passage2
Proposed PRF	0.3242	0.0212	0.2040	0.0260
SVM	0.3435*	0.0249*	0.2527*	0.0306*
MART	0.3434*	0.0247*	0.2505*	0.0308*
RankNet	0.3420*	0.0236*	0.2432*	0.0292*
RankBoost	**0.3452***†	**0.0251***†	0.2523*	0.0309*†
ListNet	0.3316*	0.0234*	0.2256*	0.0290*
LambdaMART	0.3439*	0.0250*†	**0.2540***†	**0.0309***†

'' indicates significant improvements over the proposed PRF method, and '†' indicates significant improvements over the SVM-based model.*

TABLE 6
Retrieval Performance with All the Features on the 2007 Queries

Methods	Document	Passage	Aspect	Passage2
Proposed PRF	0.2818	0.0706	0.1996	0.0992
SVM	0.3185*	0.0809*	0.2639*	0.1112*
MART	0.3140*	0.0816*†	0.2589*	0.1111*
RankNet	0.2997*	0.0769*	0.2365*	0.1070*
RankBoost	**0.3293***†	0.0832*†	**0.2685***†	0.1153*†
ListNet	0.2819*	0.0739*	0.2255*	0.1012*
LambdaMART	0.3273*†	**0.0850***†	0.2638*	**0.1163***†

'' indicates significant improvements over the proposed PRF method, and '†' indicates significant improvements over the SVM-based model.*

Table 5 shows that all the experimental results based on term-ranking models outperform the proposed PRF method, which demonstrates the usefulness of term ranking in choosing more effective expansion terms. In comparing different term-ranking models, we find that the RankNet-based and ListNet-based ranking models achieve relatively lower improvements than the other term-ranking models, and the RankBoost-based and LambdaMART-based ranking models achieve the best retrieval performance, which is analogous to the results obtained based on term-ranking accuracies. The term rankings produced by RankNet and ListNet achieve lower ranking accuracies and, consequently, contribute less to retrieval performance. Similarly, the term rankings produced by RankBoost and LambdaMART achieve higher term-ranking accuracies, and thus, these two models can help choose more useful expansion terms to improve retrieval performance. Table 6 shows trends similar to those presented in Table 5; namely, the RankBoost-based and LambdaMART-based ranking models achieve the best retrieval performance. Furthermore, we examine the retrieval performance of term-ranking models based on different feature sets. The experimental results are shown in Figs. 5 and 6.

Fig. 5 shows that the term-ranking models based on the resource-based features achieve better retrieval performance than those based on the corpus-based features, and the term-ranking models based on all the features achieve the best performance for most learning-to-rank methods except for RankNet and ListNet, which is consistent with the results obtained for the term-ranking accuracies.

Fig. 6 shows differences in the performance of the SVM-based and RankBoost-based ranking models. For these two

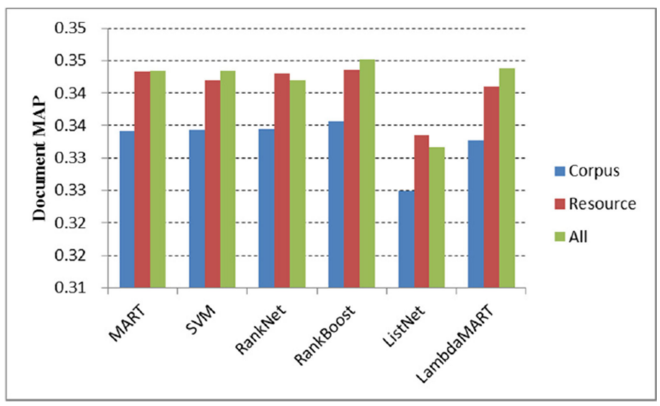


Fig. 5. Retrieval performance of different feature sets on the 2006 queries.

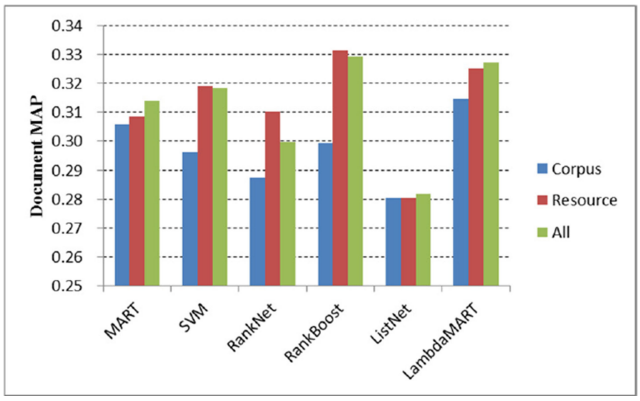


Fig. 6. Retrieval performance of different feature sets on the 2007 queries.

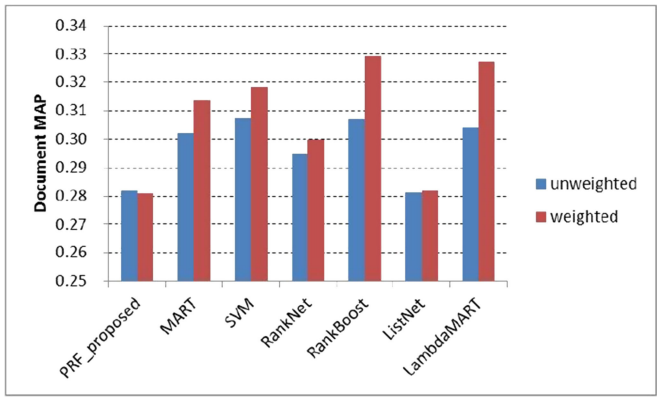


Fig. 8. Retrieval performance on term weighting for the 2007 queries.

methods, with respect to the 2007 queries, the term-ranking models based on the resource-based features outperform those based on all the features. Overall, the term-ranking models based on learning-to-rank methods with both the corpus-based features and the resource-based features can help choose more useful expansion terms for biomedical information retrieval and further improve retrieval performance.

4.5 Evaluations with Respect to Term Weighting

Because the experimental results show that term weights based on the PRF scores are of little use in improving retrieval performance, in this section, we examine the influence of term weighting based on learning-to-rank based scores on retrieval performance in terms of Document MAP. We report the experimental results in Figs. 7 and 8.

The figures show that weighted expanded queries produce better results than unweighted ones for term-ranking models based on all the learning-to-rank methods, which indicates that the learning-to-rank based scores are more effective for term weighting in the expanded query. One reason for this phenomenon may be that PRF-based scores are designed to choose candidate expansion terms, which vary greatly among different expansion terms, thus making the scores an inaccurate indicator of term importance as term weights in the expanded queries. On the other hand, ranking-based scores are designed to obtain a ranking list of all the expansion terms, taking a set of expansion terms as a whole to compute the final scores, and the scores among different expansion terms are comparable to each other.

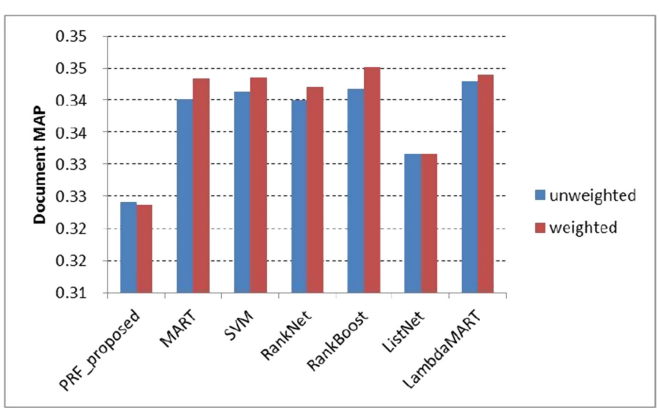


Fig. 7. Retrieval performance on term weighting for the 2006 queries.

Therefore, the ranking-based scores of terms are more suitable for use as the term weights to indicate the importance of different terms in the expanded query.

Therefore, we believe that learning-to-rank methods have two advantages in our query expansion framework for term refinement. First, learning-to-rank methods can choose more relevant expansion terms to construct the expanded query. Second, the ranking-based scores obtained using the trained models can be a more accurate indicator of term relevance for term weighting in the expanded queries to further improve retrieval performance.

4.6 Comparisons with Respect to Official Runs in TREC Genomics Tracks

To further evaluate our retrieval performance, we compare our results with the median results, the mean results and the best results reported in the 2006 and 2007 tracks of TREC Genomics [40]. The results of the comparisons are presented in Table 7.

The table shows that the proposed methods based on RankBoost and LambdaMART largely improve the mean result and the median result in the 2006 track in terms of Document MAP and Aspect MAP and outperform the best results in the 2007 track in terms of most evaluation measures. The comparison indicates that our method achieves better results in the 2007 track. For different evaluation measures, our method achieves better results in terms of Document MAP and Aspect MAP but does not perform as well in terms of Passage MAP or Passage2 MAP. One

TABLE 7
Comparison with the Best and Mean Results in the Genomics Tracks

For the 2006 queries	Document	Passage	Aspect	Passage2
Median MAP	0.3083	0.0316	0.1581	0.0345
Mean MAP	0.2887	0.0347	0.1643	0.0392
Best MAP	**0.5439**	**0.1012**	**0.4411**	**0.1486**
RankBoost	0.3452	0.0251	0.2523	0.0309
LambdaMART	0.3439	0.0250	0.2540	0.0309
For the 2007 queries	**Document**	**Passage**	**Aspect**	**Passage2**
Median MAP	0.1897	0.0565	0.1311	0.0377
Mean MAP	0.1862	0.0560	0.1326	0.0398
Best MAP	0.3286	**0.0976**	0.2631	0.1148
RankBoost	**0.3293**	0.0832	**0.2685**	0.1153
LambdaMART	0.3273	0.0850	0.2638	**0.1163**

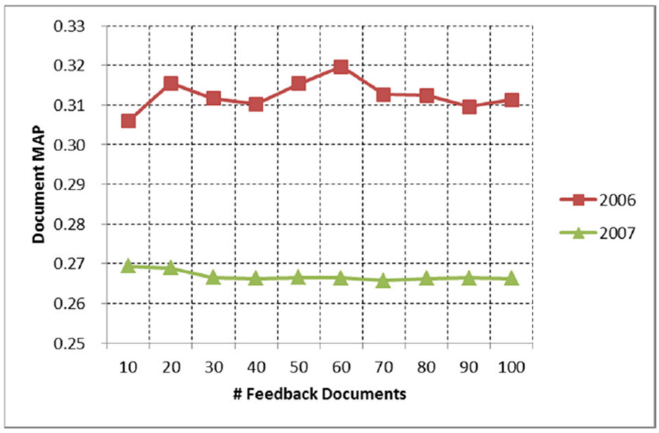


Fig. 9. Sensitivity of number of feedback documents for 2006 queries and 2007 queries of TREC Genomics Track Datasets.

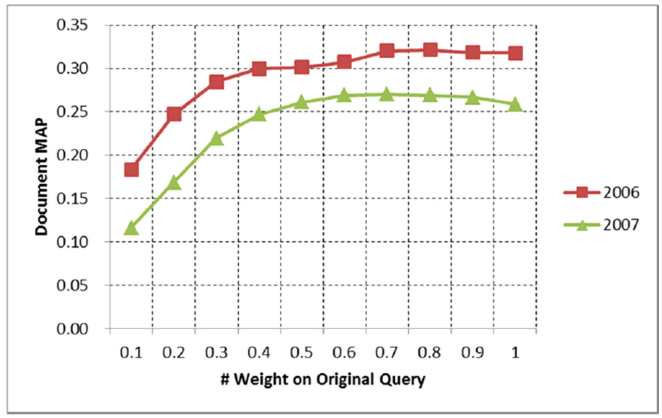


Fig. 11. Sensitivity of weight on original query for 2006 queries and 2007 queries of TREC Genomics Track Datasets.

explanation for this finding may be that Passage MAP and Passage2 MAP compute individual levels of precision based on character-level precision, which requires extra processing after retrieval by further splitting the retrieved passages into relevant pieces. Because our method does not seek to optimize this step, our results indicate lower performance based on these measures, which can be the subject of future work on further optimizing our method.

By further comparing the results for the 2006 queries and the 2007 queries, we find that our models achieve better results for the 2007 queries, which may be caused by the difference between these two query sets. Compared with the 2006 track, the 2007 track designated certain types of required entities to each query [40], such as genes, proteins, diseases and mutations, which may help better interpret information needs and benefit the construction of term-ranking models. Therefore, better results are obtained for the 2007 queries.

4.7 Parameter Selection and Discussion

In this section, we investigate the sensitivity of the parameters of pseudo-relevance feedback in our framework to retrieval performance and provide further analysis and discussions of the experimental results. There are four parameters in our query expansion framework: the number of feedback documents, the number of expansion terms, the weighting ratio on the original query and the interpolation parameter λ in the proposed PRF method. We tune the parameters for the 2007 queries with the 2006 queries and tune the parameters for the 2006 queries with the 2007 queries. The parameter tuning is illustrated in Fig. 9, 10, 11, and 12.

The figures show that for the 2006 queries we achieve the best performance when setting the number of feedback documents to 60, the number of expansion terms to 30, the weighting on the original query to 0.8 and λ to 0.3. For the 2007 queries, we find that the best performance can be achieved when setting the number of feedback documents to 10, the number of expansion terms to 30, the weighting on the original query to 0.7 and λ to 0.6.

The experimental results indicate that term-ranking accuracy does correlate with actual retrieval performance for both datasets. Thus, a higher term-ranking accuracy would contribute more to retrieval performance. A possible explanation for this finding may be that accurate term ranking can rank more relevant expansion terms highly on the term-ranking list, and more relevant expansion terms could contribute to reformulating a high-quality expanded query, leading to better retrieval performance. Moreover, accurate term ranking produces accurate weights on the chosen expansion terms, which would further improve retrieval performance.

In our query expansion framework, we introduce learning-to-rank methods for term refinement. Because different learning-to-rank methods employ different machine learning methods to train the ranking models, the

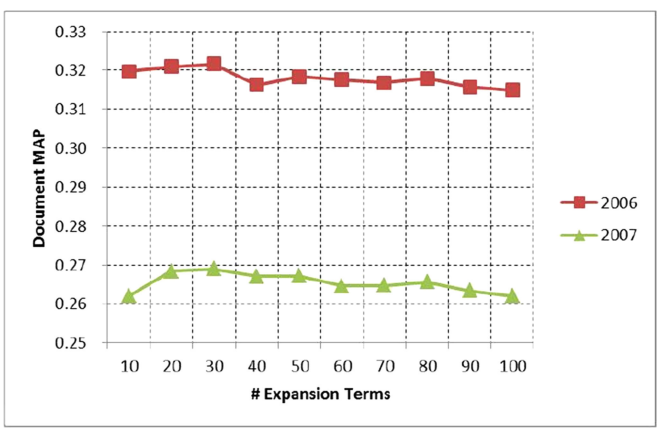


Fig. 10. Sensitivity of number of expansion terms for 2006 queries and 2007 queries of TREC Genomics Track Datasets.

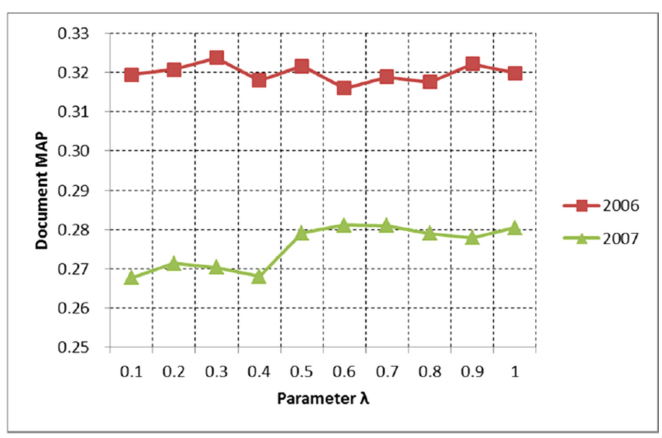


Fig. 12. Sensitivity of parameter λ for 2006 queries and 2007 queries of TREC Genomics Track Datasets.

experimental results suggest that neural network-based methods, such as RankNet and ListNet, provide a lower degree of improvement in retrieval performance although these methods achieve better performance when training only with the resource-based features. Boosting- and regression tree-based methods, such as MART, RankBoost and LambdaMART, achieve better performance than other methods. For the features used in the training phase, the resource-based features can indeed enhance the term-ranking models when considering domain-specific characteristics for better term refinement, and we can enhance retrieval performance further by using both the resource-based features and the corpus-based features.

Overall, we attribute the improvement in biomedical information retrieval of the proposed framework to four aspects, namely, the candidate expansion term extraction, the term-labeling strategy, the term features and the ranking models. Our experiments also show that these aspects contribute to improving retrieval performance. For the candidate expansion term extraction, we combine the MeSH-based information with the co-occurrence-based information, which helps to choose a large set of terms covering more relevant terms for further refinement. For the term-labeling strategy, we propose integrating the increasing magnitude of the performance into measuring the relevance of candidate terms, which yields more accurate labels and contributes to more effective ranking models. For the term features, we extract both the corpus-based features and the resource-based features, which depict the usefulness of terms more completely from different perspectives and complement each other in constructing the term-ranking models. For the ranking models, we investigate three approaches to learning to rank in our experiments, and find that the LambdaMART-based term ranking can achieve the best performance. These four aspects jointly contribute to the improvement in retrieval performance, and the proposed framework can also be further optimized in these respects to enhance biomedical retrieval performance.

5 Conclusions and Future Work

In this paper, we propose a novel query expansion framework based on learning-to-rank methods for biomedical information retrieval. In the framework, we propose incorporating the MeSH thesaurus into the co-occurrence-based term selection method to select the candidate expansion terms. To refine the expansion terms, we define and extract both the corpus-based term features and the resource-based term features to represent the terms as feature vectors, which are taken as the inputs for learning-to-rank methods to learn the term-ranking models. Different approaches to learning-to-rank are investigated in our framework for training the term-ranking models. The proposed framework for biomedical query expansion based on learning-to-rank not only contributes to choosing relevant terms for high-quality expanded queries, but also yields effective term weights to further enhance retrieval performance. Experimental results obtained for TREC Genomics datasets show that our framework can better refine the expansion terms and improve the performance of biomedical information retrieval.

Our future work will be extended in two directions. On one hand, we will explore other biomedical resources for extracting powerful term features, which can be of great use for optimizing expansion term refinement. On the other hand, because domain-specific concepts are important in biomedical information retrieval, we will attempt to construct biomedical concept ranking models to further improve our query expansion framework.

Acknowledgments

This work is partially supported by grants from the Natural Science Foundation of China (No. 61632011, 61572102, 61751203, 61602078, 61402075), the Fundamental Research Funds for the Central Universities (DUT16ZD216), and the National Key Research Development Program of China (No. 2016YFB1001103). The authors would like to express their sincere appreciations to editors and anonymous reviewers for their invaluable comments in improving their manuscript.

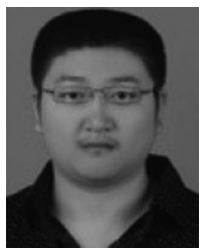
References

[1] J. Rocchio, "Relevance feedback in information retrieval," in *the Smart Retrieval System*, Englewood Cliffs, NJ, USA: Prentice Hall, 1971, pp. 313–323.
[2] V. Lavrenko and W. B. Croft, "Relevance based language models," in *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2001, pp. 120–127.
[3] S.E. Robertson, S. Walker, M. Beaulieu, M. Gatford, and A. Payne, "Okapi at TREC-4," in *Proc. 4th Text Retrieval Conf. NIST Special Publication*, 1996, pp. 73–97.
[4] C. Zhai and J. Lafferty, "Model-based feedback in the language modeling approach to information retrieval," in *Proc. 10th Int. Conf. Inf. Knowl. Manage.*, 2001, pp. 403–410.
[5] C. J. Lee, R. C. Chen, S. H. Kao, and P. J. Cheng, "A term dependency-based approach for query terms ranking," in *Proc. 18st ACM Int. Conf. Inf. Knowl. Manage.*, 2009, pp. 1267–1276.
[6] G. Cao, J. Y. Nie, J. Gao, and S. Robertson, "Selecting good expansion terms for pseudo-relevance feedback," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 243–250.
[7] Y. Lin, H. Lin, S. Jin, and Z. Ye, "Social annotation in query expansion: A machine learning approach," in *Proc. 34st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 405–414.
[8] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, 2011, pp. 1189–1232.
[9] C. Burges, et al., "Learning to rank using gradient descent," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 89–96.
[10] Z. Cao, T. Qin, T. Y. Liu, M. F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 129–136.
[11] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *J. Mach. Learn. Res.*, vol. 4, pp. 933–969, 2003.
[12] C. J. Burges, "From RankNet to LambdaRank to LambdaMART: An overview," *Learn.*, vol. 11, pp. 23–581, 2010.
[13] C. Quoc and V. Le, "Learning to rank with nonsmooth cost functions," in *Proc. Advances Neural Inf. Process. Syst.*, vol. 19, pp. 193–200, 2007.
[14] T. Y. Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.
[15] T. Qin, T. Y. Liu, J. Xu, and H. Li, "LETOR: A benchmark collection for research on learning to rank for information retrieval," *Inf. Retrieval*, vol. 13, no. 4, pp. 346–374, 2010.
[16] Z. Ji and B. Wang, "Learning to rank for question routing in community question answering," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 2363–2368.
[17] J. Sun, S. Wang, B. J. Gao, and J. Ma, "Learning to rank for hybrid recommendation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 2239–2242.
[18] B. Xu, H. Lin, and Y. Lin, "Assessment of learning to rank methods for query expansion," *J. Assoc. Inf. Sci. Technol.*, vol. 67, no. 6, pp. 1345–1357, 2016.
[19] P. Srinivasan, "Query expansion and MEDLINE," *Inf. Process. Manage.*, vol. 32, no. 4, pp. 431–443, 1996.

[20] X. Xu, W. Zhu, X. Zhang, X. Hu, and I. Y. Song, "A comparison of local analysis, global analysis and ontology-based query expansion strategies for biomedical literature search," in *IEEE Int. Conf. Syst. Man Cybern.*, 2006, vol. 4, pp. 3441–3446.
[21] S. Matos, J. P. Arrais, J. Maia-Rodrigues, and J. L. Oliveira, "Concept-based query expansion for retrieving gene related publications from MEDLINE," *BMC Bioinf.*, vol. 11, no. 1, 2010, Art. no. 212.
[22] A. R. Rivas, E. L Iglesias, and L. Borrajo, "Study of query expansion techniques and their application in the biomedical information retrieval," *Sci. World J.*, vol. 2014, 2014, Art. no. 132158.
[23] D. Metzler and W. B. Croft, "Linear feature-based models for information retrieval," *Inf. Retrieval*, vol. 10, no. 3, pp. 257–274, 2007.
[24] M. Bendersky, D. Metzler, and W. B. Croft, "Learning concept importance using a weighted dependence model," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 31–40.
[25] M. Ellen Voorhees and M. Richard, "Overview of the TREC 2011 medical records track," in *Proc. Text Retrieval Conf.*, 2011.
[26] E. M. Voorhees and W. R. Hersh, "Overview of the TREC 2012 medical records track," in *Proc. Text Retrieval Conf.*, 2012.
[27] Y. Wang, X. Liu, and H. Fang, "A study of concept-based weighting regularization for medical records search," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 603–612.
[28] D. Zhu, S. Wu, B. Carterette, and H. Liu, "Using large clinical corpora for query expansion in text-based cohort identification," *J. Biomed. Inf.*, vol. 49, pp. 275–281, 2014.
[29] S.T. Wu, D. Zhu, W. Hersh, and H. Liu, "Clinical information retrieval with split-layer language models," in *Proc. ACM SIGIR Workshop Health Search Discovery*, 2013, p. 51.
[30] K. Dramé, F. Mougin, and G. Diallo, "Query expansion using external resources for improving information retrieval in the biomedical domain," in *Proc. Workshop CLEF (Working Notes)*, 2014, pp. 189–194.
[31] H. S. Oh and Y. Jung, "Cluster-based query expansion using external collections in medical information retrieval," *J. Biomed. Informat.*, vol. 58, pp. 70–79, 2015.
[32] J. Mao, K. Lu, X. Mu, and G. Li, "Mining document, concept, and term associations for effective biomedical retrieval: Introducing MeSH-enhanced retrieval models," *Inf. Retrieval J.*, vol. 18, no. 5, pp. 413–444, 2015.
[33] V. Jalali and M. R. M Borujerdi, "The effect of using domain specific ontologies in query expansion in medical field," in *Proc. Int. Conf. Innovations Inf. Technol.*, 2008, pp. 277–281.
[34] B. Xu, et al., "Improve Biomedical Information Retrieval using Modified Learning to Rank Methods," *IEEE/ACM Trans. Comput. Biology Bioinf.*, vol. 15, no. 6, pp. 1797–1809, Nov./Dec. 2018.
[35] J. M. Ponte and W.B. Croft, "A language modeling approach to information retrieval," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1998, pp. 275–281.
[36] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft, "Indri: A language model-based search engine for complex queries," in *Proc. Int. Conf. Intell. Anal.*, 2004, pp. 2–6.
[37] M. Lease, J. Allan, and W. B. Croft, "Regression rank: Learning to meet the opportunity of descriptive queries," in *Advances in Information Retrieval*, Berlin, Germany: Springer, pp. 90–101, 2009.
[38] D. Zhu and B. Carterette, "An adaptive evidence weighting method for medical record search," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 1025–1028.
[39] A. R. Aronson, "Effective mapping of biomedical text to the UMLS metathesaurus: The MetaMap program," in *Proc. AMIA Symp.*, 2001, Art. no. 17.
[40] W. Hersh and E. Voorhees, "TREC genomics special issue overview," *Inf. Retrieval*, vol. 12, no. 1, pp. 1–15, 2009.
[41] W. Hersh, A. M. Cohen, P. M. Roberts, and H. K. Rekapalli, "TREC 2006 genomics track overview," in *Proc. Text Retrieval Conf.*, 2006, pp. 14–23.
[42] D. Metzler and W.B. Croft, "A markov random field model for term dependencies," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2005, pp. 472–479.

Bo Xu received the BSc degrees from the Dalian University of Technology, China, in 2011. He is currently working towards the PhD degree in the School of Computer Science and Technology, Dalian University of Technology. His current research interests include information retrieval, learning to rank, natural language processing, and biomedical text mining.

Hongfei Lin received the BSc degree from Northeastern Normal University, in 1983, the MSc degree from the Dalian University of Technology, in 1992, and the PhD degree from Northeastern University, in 2000. He is currently a professor in the School of Computer Science and Technology, Dalian University of Technology. He has published more than 100 research papers in various journals, conferences, and books. His research interests include information retrieval, text mining, natural language processing, and effective computing. In recent years, he has focused on text mining for biomedical literatures, biomedical hypothesis generation, information extraction from huge biomedical resources, learning to rank, sentimental analysis, and opinion mining. He is the director of the Information Retrieval Laboratory, Dalian University of Technology. His research projects are funded by the National Natural Science Foundation of China and National High-Tech Development Plan.



Yuan Lin received the BSc and PhD degrees from the Dalian University of Technology, China, in 2006 and 2012, respectively. He is currently a lecturer in the School of Public Administration and Law, Dalian University of Technology. His current research interests include information retrieval and learning to rank.