Improve Biomedical Information Retrieval Using Modified Learning to Rank Methods

Bo Xu, Hongfei Lin, Yuan Lin, Yunlong Ma, Liang Yang, Jian Wang, and Zhihao Yang

Abstract—In these years, the number of biomedical articles has increased exponentially, which becomes a problem for biologists to capture all the needed information manually. Information retrieval technologies, as the core of search engines, can deal with the problem automatically, providing users with the needed information. However, it is a great challenge to apply these technologies directly for biomedical retrieval, because of the abundance of domain specific terminologies. To enhance biomedical retrieval, we propose a novel framework based on learning to rank. Learning to rank is a series of state-of-the-art information retrieval techniques, and has been proved effective in many information retrieval tasks. In the proposed framework, we attempt to tackle the problem of the abundance of terminologies by constructing ranking models, which focus on not only retrieving the most relevant documents, but also diversifying the searching results to increase the completeness of the resulting list for a given query. In the model training, we propose two novel document labeling strategies, and combine several traditional retrieval models as learning features. Besides, we also investigate the usefulness of different learning to rank approaches in our framework. Experimental results on TREC Genomics datasets demonstrate the effectiveness of our framework for biomedical information retrieval.

Index Terms—Information retrieval, machine learning, supervised learning, text mining

1 Introduction

In recent years, research articles in biomedicine domain have increased exponentially, which makes it difficult for biologists to manually capture all the information they need. To meet biologists' information need better, information retrieval (IR) techniques designed for biomedicine domain have been addressed, focusing on how to effectively retrieve the needed information. Given a query, an IR system can search for its relevant documents, and rank the documents based on their relevance degrees to the query. Unlike traditional IR, biomedical IR faces some domain specific challenges, most of which are due to the abundance of the terminologies. Different articles may use different terminologies to represent the same concept, and as a result, two relevant documents for the same query may vary a lot. To meet the information need more completely, biomedical IR system should cover the relevance documents from different aspects, where an aspect of relevance documents refers to a subset of relevant documents related to the same terminologies.

Therefore, biomedical retrieval systems not only focus on obtaining the most relevant documents to a given query, but also emphasize the query-related aspects coverage in the document ranked list, which is mostly denoted as the diversity of the searching result. To explain the diversity further, consider an example of a biomedical query "What is the role of MMS2 (a kind of enzyme) in cancer?". Given the query, we retrieve its relevant documents based on their closeness to the query, and then relevant documents are divided into multiple groups. Each group has a unique label, such as "cell differentiation", "DNA repair", "DNA damage", each reflecting one aspect of the query. All the query aspects covered in the document ranking list indicate the diversity degree of the results. Our searching goal is to retrieve the most relevant documents covering as many aspects as possible.

In recent years, various traditional IR models have been introduced for the biomedical document ranking, and achieve some good results. Learning to rank, as a state-of-the-art IR technique, has been proved effective in many IR tasks, which solves ranking problems using machine learning methods with various features, and many learning to rank methods have been proposed [1], [2], [3], [4], [5], [6], [7], [8]. However, few studies attempted to employ learning to rank methods to improve the diversity oriented biomedical information retrieval. Learning to rank methods have some advantages over other traditional IR models. For one thing, it can make the most of various ranking information comprehensively to construct a ranking model. The ranking information can be either statistical textual features, such as term frequency, or some features obtained from traditional IR models, such as vector space model. For another, the training phase for learning to rank methods iteratively reduce the value of ranking loss (i.e., the difference between the predicted ranking and the ground truth ranking) until eventually an optimal ranking model is obtained. Therefore, it seems promising to improve biomedical retrieval using learning to rank methods.

In the paper, we propose a novel framework based on learning to rank methods to study whether learning to rank

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

Manuscript received 4 Feb. 2016; revised 18 Apr. 2016; accepted 31 May 2016. Date of publication 14 June 2016; date of current version 6 Dec. 2018.
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.2016.2578337

methods would benefit biomedical retrieval and boost both the relevance and the diversity of results. In the framework, we propose two novel labeling strategies to capture the aspect information of the relevant documents, thus forming the ground truth document labels. Meanwhile, we represent the documents to a given query as feature vectors by scoring the documents using different traditional IR models. Then, we construct an effective ranking model using these feature vectors as training data to improve retrieval performance. Finally, for a new query, we predict its corresponding document ranking using the trained model.

The contributions of the paper are listed as follows:

1) we propose a learning framework to integrate learning to rank methods into biomedical information retrieval, and compare the performance of several state-of-the-art learning to rank methods in this framework;
2) we propose two labeling strategies: one is focusing on constructing an optimal ranking target, and the other is based on the group-wise learning to rank method;
3) we examine the effectiveness of the proposed framework on TREC Genomics Track datasets, and compare the performance of different learning to rank methods for our framework.

The remainder of our paper is organized as follows: In Section 2, we review the existing literature, and contrast our method with the related work; in Section 3, we describe the proposed method in more details; in Section 4, we present our experimental results and give some analysis on the results; we conclude this paper and discuss some future directions in Section 5.

2 Related Work

In this section, we will review the related work in three lines: biomedical information retrieval, information retrieval for diversity and learning to rank methods.

2.1 Biomedical Information Retrieval

In biomedical information retrieval, ranking only based on document relevance is not sufficient to meet the information need, because relevant documents may be redundant with each other. Aspect retrieval was proposed to reduce the redundancy and improve result diversity in the Genomics track of Text Retrieval Conference (TREC) [9], [10]. Diversity here means that when a user submits a query to a retrieval system, he (or she) is provided with the diversified results covering as many aspects of the query as possible, and then the user will find what he desires the most.

In the 2006 TREC Genomics track, University of Wisconsin at Madison proposed a clustering approach, but failed to promote diversity by penalizing redundancy [11]. In the 2007 TREC Genomics track, most submissions are purely based on relevance passage retrieval such as National Library of Medicine (NLM) [12]. Thereafter, some researches focused on modeling the diversity by detecting query-related potential aspects. Yin, Huang, and Li [13] utilized Wikipedia to detect aspects, and proposed a cost based document re-ranking method to balance the relevance and the diversity of retrieval performance. Based on Wikipedia aspect detection, a survival modeling method was introduced to model the passage diversity [14], and a relevance-novelty model, RelNov, was proposed to improve passage retrieval [15]. In [16], a topic modeling method based on Latent Dirichlet Allocation (LDA) is proposed to measure the novelty of a given passage. In [17], a retrieval model based on Probabilistic Latent Semantic Analysis (PLSA) was proposed to detect latent aspects for diversity retrieval. However, few studies focused on improving ranking performance using supervised learning methods, such as learning to rank methods, which have been demonstrated powerful in traditional IR. The challenge lies in how to adapt these methods to biomedical IR.

2.2 Information Retrieval for Diversity

Information retrieval for diversity is a retrieval task that diversifies the search results to meet the multiple information needs of different users on the basis of traditional information retrieval [32], [33]. In order to meet the information needs, top-ranked documents in search results should not only contain as many relevant documents as possible, but also cover as many query-related aspects as possible, in which the relevant documents or related aspects are judged by human experts in advance.

Most existing diversity orient retrieval methods can be divided into two categories: implicit approaches and explicit approaches. Implicit approaches model query-related aspects by modeling the relationship among documents. For example, Zhai, Cohen and Lafferty [34] rank high diversified documents by differentiate the divergence of their language models. On the other hand, explicit approaches model the query-related aspects using external resources, such as the top-ranked documents [35], Wikipedia [13] and taxonomy [32], [36]. In comparison, explicit approaches outperform implicit ones in most circumstances. Different from these methods, we improve the diversity of search results in a supervised way by explicitly using the aspect information to train diversity oriented ranking models.

Recently, some research improves the diversity using machine learning methods. For example, Yue and Joachims [37] take the subtopic coverage as loss function to optimize the results list. However, they ignore the relevance degree of results while increasing the diversity degree. In this paper, we propose a learning framework to improve both the diversity and the relevance of search results based on learning to rank, which adopts machine learning methods in information retrieval, and has been proved effective in many tasks.

2.3 Learning to Rank Methods

In IR, learning to rank, as a powerful technology, has been proved effective in improving relevant-based retrieval performance in the intersection of machine learning and information retrieval [6], [18]. Learning to rank for IR is a task to automatically construct a ranking model, so that the model can sort new documents according to their degrees of relevance, preference, or importance with various features.

In fact, ranking can be considered as a task to choose the optimal objects from a large set of objects. Therefore, many tasks can be ultimately categorized into ranking problems and have been solved using learning to rank methods [6], [18]. For example, Ji and Wang [19] proposed a learning to rank framework for Community Question Answering (CQA) to capture the intrinsic relationships between the asker and the answerers, and Sun, Wang, Gao, et al. [20] integrated learning to rank to recommender systems to provide users with appropriate items by ranking the preference information.

In order to solve ranking problem, many learning to rank methods have been proposed to improve ranking accuracies. In particular, learning to rank is grouped into three approaches: The pointwise approach, the pairwise approach and the listwise approach. Different approaches model the learning to rank process in different ways.

Pointwise approach is a straightforward way for using machine learning technologies to solve ranking problem. When ranking with pointwise methods, one assumes that the exact relevance degree of each document is what we are going to predict, even though it may not be necessary when the target is to produce a ranked list of documents. Pairwise approach does not focus on accurately predicting the relevance degree of each document; instead, it cares about the relative order of two documents. In this sense, it is closer to the concept of ranking than pointwise approach. Listwise approach takes the document list as the object to calculate the difference between the predicted ranking list and the target ranking list of documents. Intuitively, listwise approach utilizes the most ranking information to construct the ranking model.

Besides, Lin, et al. [21] proposed group-wise learning to rank framework, and demonstrated its effectiveness. However, to the best of our knowledge, few studies attempt to introduce learning to rank methods for biomedical retrieval. In the paper, we attempt to investigate the effectiveness of learning to rank methods for biomedical information retrieval.

3 Methods

3.1 General Learning Framework

In this section, we will formalize our learning to rank based framework for biomedical retrieval. At the training time, we are given a set of N queries $Q = \{q_1, q_2, \dots, q_N\}$. To simplify notation, we drop the query index, and refer to a general query q. Each query q is associated with a set of M documents $D = \{d_1, d_2, \dots, d_M\}$. The documents are manually labeled with relevance labels, denoted as $L = \{l_1, l_2, \dots, l_M\}$. For each document d_j, label l_j is an integer indicating the relevant degree of the document d_j to the query q. In addition, each document d_j is represented as a query dependent feature vector, where $f_j[k]$ denotes the kth feature value for the document d_j. The learning goal is to create a scoring function F such that, given a set of documents D with relevance labels L for a query q, the ranking of documents in D produced by F has maximal agreement with L. Then, the scoring function, as the ranking model, is used to rank documents for new queries.

In order to adapt learning to rank framework to biomedicine domain, addressing both the relevance and the diversity of retrieval results, we modify the framework to optimize the training process, focusing on obtaining the relevant documents with the most useful query-related aspects. We choose the scores obtained from some classic IR models as learning features to capture the document information from different query related aspects. The final ranking model seeks to rank the most relevant and diversified results at the top of the ranking list. Details about our methods will be given in the following sections.

3.2 Biomedical Document Labeling

In the training phase, learning to rank methods can reduce the ranking loss by measuring the difference between the outputs and the ground truths. The ground truths refer to the relevance labels of the documents, and can be considered as the learning target to train a ranking model. In general, the ground truth label for a document is an integer, indicating the relevance degree of the document. Specifically, a document labeled as 1 is more relevant than another document labeled as 0 for the same query, where label 0 indicates the document is irrelevant to the query.

Algorithm 1. Optimal Ranking Labeling Strategy

```
Input:
  Relevant document set $R = \{r_1, r_2, \dots, r_n\}$, aspect set $Asp_{ri}$
  for each document $r_i$, the whole aspect set $Asp$
Output:
  Document labels $L = \{l_1, l_2, \dots, l_n\}$
Description:
  1: initialize $label = |R|$
  2: find documents $S$ with maximum aspects in $R$
  3: for each document $r_i$ in $S$, compute
       $\sum_{aspect_j \in Asp_{r_i}} df(aspect_j)$
  4: choose the document $r_k$ with the minimum
       $\sum_{aspect_j \in Asp_{r_i}} df(aspect_j)$
  5: $l_k = label$
  6: $label = label - 1$
  7: update $Asp$ by removing the aspects in $r_k$
  8: update $R$ by removing the document $r_k$
  Repeat step 2 to step 8 until $Asp$ is empty
  9: for the remaining documents in $R$, choose the document $r_k$
     with minimum $\sum_{aspect_j \in Asp_{r_i}} df(aspect_j)$
  10: $l_k = label$
  11: $label = label - 1$
  12: update $R$ by removing the document $r_k$
  Repeat step 9 to 11 until $R$ is empty
```

In biomedical IR, relevant documents are not only judged with relevance labels, but also explicitly annotated with some biomedical terms, and each term stands for one aspect to the query. All the aspects for one query indicate the completeness of the searching results, and the distribution of aspects in the document ranking list reflects the diversity of the ranking performance. Therefore, our task is how to generate effective document labels involving both the original relevance label and the aspect information. Based on the idea, we propose two novel labeling strategies to tackle the problem.

3.2.1 Optimal Ranking Labeling Strategy

Firstly, we attempt to construct an optimal ranking list of relevant documents in consideration of their diversity degrees. In the optimal ranking list, more diversified relevant documents are ranked higher than less diversified ones. At the training time, learning to rank methods compute the ranking loss by measuring the difference between the optimal ranking list and the predicted ranking list by the model, and then iteratively adjust the model to reduce the loss continuously. Our first labeling strategy is based on this idea by taking the number of aspects for one document and the frequency of aspects among all the documents into account, where the aspects for a relevant document reflect its diversity degree. The algorithm is shown in Algorithm 1.

In this algorithm, $df(aspect_j)$ counts the number of relevant documents covering the *j*th aspect. Given a query, there are a set of relevant documents $R = \{r_1, r_2, \ldots, r_n\}$. One relevant document may cover several query related aspects, and one aspect may be shared by many relevant documents. Therefore, we take these two factors into account to form the ground truth labels of the documents. The final ground truth labels for relevant documents are integers ranging from 1, 2 to *n*, indicating the diversity degrees of these documents from low to high, where *n* represents the total number of relevant documents. Besides, irrelevant documents are labeled as 0. For example, if there are five relevant documents for a given query, we will respectively label these documents as 1, 2, 3, 4 and 5.

Specifically, there are two phases in this algorithm. In the first phase, we choose the candidate set of documents covering most aspects, and sum up the occurrences of the aspects related to each document. Then, we choose the document with fewest occurrences of aspects, because the document with fewer occurrences is more preferable to rank higher. We repeat this step until all the query-related aspects are counted. In the second phase, we rank the remaining aspects based on the summation of their occurrences, and give them labels from high to low. In this way, we finally label the documents based on both relevance and aspect information.

3.2.2 Group-Wise Labeling Strategy

Optimal ranking strategy provides the target ranking to train the ranking models, which may be more suitable for listwise learning to rank approach, because it directly measures the difference between the target ranking list and the predicted ranking list. However, for pointwise and pairwise learning to rank methods, it may not work well, because they respectively utilize the exact relevance degree of each document and preferences between two documents to compute the ranking loss. Based on this consideration, we propose another labeling strategy to examine learning to rank methods.

Inspired by the group-wise learning to rank framework proposed in [21], we propose a diversity-oriented group-wise learning to rank framework to improve the retrieval diversity. In this framework, documents with different labels are treated as a group, and the ranking task is then reduced from ranking the whole set of documents to ranking a group of documents with different labels.

Before presenting the labeling strategy, we briefly introduce the group-wise learning to rank [21]. In the first presentation of group-wise learning to rank, the training set is divided into groups. Each group contains one relevant document and several irrelevant documents, and all the groups constitute the whole training set. In essence, group-wise learning to rank employs the methodology of divide-and-conquer, to train ranking models with smaller units instead of taking all the documents as a whole. We modify the group-wise learning to rank framework to make it fit into the biomedical diversity-oriented retrieval. This algorithm is presented in Algorithm 2.

Algorithm 2. Optimal Ranking Labeling Strategy

```
Input:
  Relevant document set $R = \{r_1, r_2, \ldots, r_n\}$, aspect set $Asp_{ri}$ for each document $r_i$
Output:
  Document labels $L = \{l_1, l_2, \ldots, l_n\}$, document groups $G = \{g_1, g_2, \ldots, g_n\}$
Description:
  1: find documents S with maximum aspects in R
  2: for each document $r_i$ in S, compute
     $\sum_{aspect_j \in Asp_{r_i}} df(aspect_j)$
  3: choose the document $r_k$ with the minimum
     $\sum_{aspect_j \in Asp_{r_i}} df(aspect_j)$
  4: $l_k = 1, g_k = k$
  5: update R by removing the document $r_k$
  6: for each document $r_i$ in R
  7:     if $Asp_{ri} = Asp_{rk}$
  8:         $l_k = 1, g_k = k$, update R by removing the document $r_k$
  9:     if $Asp_{ri} \subset Asp_{rk}$
  10:        $l_k = 0,\ g_k = k$, update R by removing the document $r_k$
  11: end for
  Repeat step 1 to step 11 until R is empty
```

In the algorithm, we firstly divide the relevant documents into groups based on their covered aspects. Each group contains one document with more aspects (label 1) and several documents with less aspects (label 0), and the document with more aspects covers all the aspects in the documents with less aspects. Besides, the documents with the same set of aspects are assigned into the same group with the same label. After dividing the relevant documents into groups, we assign each group some irrelevant documents. As a result, one or more relevant documents and a group of irrelevant documents constitute the whole of one group, which can be taken as a learning unit at the model training time. We generate the group-wise labels according to the aspect sets of documents, especially the inclusion relation between two aspect sets. The resulted groups can be considered as a complete division of all the documents, and the size of each final group is averaged over the number of documents by the number of groups.

Since the division of groups is based on the diversity degrees of the documents, the group-wise framework can be more focused on the diversified documents, and the final ranking model may improve the performance in terms of both relevance and diversity.

3.3 Ranking Features

In this section, we introduce the document features for training ranking models obtained from some classic retrieval models in IR, including the vector space model, the Okapi BM25 model and the language models. These models have been proved effective in many IR tasks, and can gain average performance for biomedical information retrieval. Different ranking features may focus on different aspects of a query. Therefore, we take these models as features in our learning to rank framework to capture the different aspects of the results.

3.3.1 Features Based on Vector Space Model

Vector space model (VSM) has been widely used in information retrieval field, which is a simple and intuitively appealing framework for implementing term weighting and ranking [22]. In VSM, terms are weighted using term frequency inverse document frequency (TF-IDF), which is a classic way to model term importance in data collections. Using the term importance, we can calculate the cosine similarity between a document *d* and a query *q* as follows:

$$\text{cosine}(d, q) = \frac{\sum_{j \in q} w_d(j) \cdot w_q(j)}{\sqrt{\sum_{j \in q} w_d^2(j) \cdot \sum_{j \in q} w_q^2(j)}} \tag{1}$$

$$\text{tf}(j, d) = \frac{occurrence_d(j)}{|d| + 1.0} \tag{2}$$

$$\text{idf}(j) = \log \frac{N - n(j) + 0.5}{n(j) + 0.5} \tag{3}$$

$$w_d(j) = \text{t}f(j, d) \cdot \text{i}df(j), \tag{4}$$

where $w_d(j)$ is the weight for query term *j* in the document *d*, and $w_q(j)$ is the weight for query term *j* in the query. $w_d(j)$ and $w_q(j)$ can be calculated using (2), (3) and (4). In (2), $occurrence_d(j)$ counts the number of occurrences of query term *j* in the document *d*. In (3), $n(j)$ is the number of documents containing term *j*, and *N* is the total number of documents in the whole collection.

3.3.2 Features Based on BM25 Model

Okapi BM25 model takes into account the document length to overcome the shortcoming of vector space model (VSM) [23], which has been proved effective in many tasks. The similarity scoring function between a query *q* and its corresponding candidate document *d* can be computed as follows:

$$\begin{aligned}\text{BM25}(d, q) = \sum\nolimits_{j \in q} \text{i}df(j) \cdot \frac{(k_3 + 1.0) \cdot \text{t}f(j, q)}{k_3 + \text{t}f(j, q)} \\ \cdot \frac{\text{t}f(j, d) \cdot (k_1 + 1.0)}{\text{t}f(j, d) + k_1 \cdot (1 - b + b \cdot |d|/avgdl)},\end{aligned} \tag{5}$$

where k_1, b, and k_3 are parameters, $|d|$ is the document length and $avgdl$ is the average document length in the whole collection. We switch the parameters of BM25 to obtain different features. Specifically, We empirically set the initial parameter $k_1 = 1.0$, $b = 0.75$ and $k_3 = 7$, and then, we fix two parameters and switch the other parameter to obtain different scores: $k_1 \text{in}\{1, 2, \ldots, 10\}, k_3 \text{in}\{1, 2, \ldots, 10\}$ and $b \,\text{in}\{0.25, 0.5, 0.75\}$. Finally, we obtain 23 features based on BM25.

3.3.3 Features Based on Language Models

The unigram language model (LM) is often used in traditional IR, assuming that each term is generated independently. It concerns the probabilities of sampling a single word by the maximum likelihood. To avoid zero probability, different smoothing methods are adopted [24], and we generate different features based on different smoothing methods to make the ranking model more reliable.

Language model with Jelinek-Mercer smoothing involves a linear interpolation of the maximum likelihood model with the collection model, and can be calculated as follows:

$$w_d(j) = (1 - \lambda) \cdot \text{t}f(j, d) + \lambda \cdot \text{t}f(j, C), \tag{6}$$

where $\text{tf}(j, C)$ is the number of occurrences of query term *j* in the whole collection *C*. We switch the parameter$\boldsymbol{\lambda}$ in the set $\{0.1, 0.2, \ldots, 1.0\}$ to obtain 11 different features. Language model with Bayesian smoothing is a multinomial distribution using Dirichlet priors, and can be calculated as follows:

$$w_d(j) = \frac{\text{t}f(j, d) + \mu \cdot \text{t}f(j, C)}{\sum_i \text{t}f(i, d) + \mu}. \tag{7}$$

Similarly, we switch the parameter μ in the set $\{1,000, 1,100, \ldots, 2,000\}$ to obtain 11 different features.

Finally, we extract 46 features in total (one feature based on VSM, 23 features based on BM25, and 22 features based on language models), which not only indicate the relevance degree of different document, but also emphasize different aspects for a given query. From this point of view, if we can make the most of these features, we will retrieve the most relevant and diversified documents to fulfill the information need.

3.4 Learning Methods

In this paper, we examine the usefulness of our framework by extending three learning to rank approaches: the pointwise approach, the pairwise approach and the listwise approach. The main difference among the approaches lies in the loss function, which is the way to compute the ranking loss.

For the pointwise approach, the ranking loss is computed based on the difference between the score obtained from the trained model and its ground truth label, and then accumulated for all the documents as the total ranking loss. Take Regression [25] as an example, its ranking loss is purely based on the square loss used in machine learning. At its model training time, it reduces the ranking loss using gradient descent in iterations until the ranking loss stops reducing or the loss reduction between two iterations is less than a fixed threshold,

$$\text{loss}(f(x_i), y_i) = \sum\nolimits_i (f(x_i) - y_i)^2. \tag{8}$$

Equation (8) is the loss function of Regression, where *f* is the ranking model, $f(x_i)$ is the predicted score of the document *i* by the model, and y_i is the ground truth label of the document using our labeling strategies. From the equation, we can see that the total loss for pointwise method is the loss summation over all the documents.

For pairwise approach, the loss function is computed based on the preferences in each document pair. For a pair of documents *i* and *j*, when *i* is preferred to *j*, the pair of documents is taken as a positive pair, and when *j* is preferred to *i*, the pair of documents is taken as a negative pair. Based on that, pairwise approach takes the number of wrongly classified positive pairs as the ranking loss.

RankBoost [4] combines preferences based on the boosting approach to machine learning. It utilizes the object pairs with preferences as instances in its training procedure, and operates in rounds by combining many weak learners, each of which is found in an iteration process and weakly correlates with the target ranking model. The final ranking model of RankBoost is an ensemble of all the weak learners,

$$\text{loss}(f(x_i), f(x_j), y_{i,j}) = \sum\nolimits_{i,j} \exp(-y_{i,j} \cdot (f(x_i) - f(x_j))). \quad (9)$$

Equation (9) is the loss function of RankBoost, where *f* is the ranking model, $f(x_i)$ and $f(x_j)$ is the predicted scores of the document *i* and the document *j*, and $y_{i,j}$ is the preference between these two documents based on the ground truth labels. From the equation, we can see that the total loss for pairwise method is the loss summation over all the document pairs.

RankNet [27] is another kind of pairwise learing to rank methods, which takes the neural network as the basic ranking model, and adopts a probabilistic loss function. RankNet firstly maps the outputs for a pair of documents to probablisties modeled by a logistic function, and then measures the ranking loss using the cross entropy between the target probability and the predicted probability. The total ranking loss accumulates the ranking loss of every pairs of documents in each training query

$$\text{loss}(Q_{ij}, P_{ij}) = \sum\nolimits_{i,j} -Q_{ij}\log P_{ij} - (1 - Q_{ij})\log(1 - P_{ij}) \quad (10)$$

$$P_{ij} = \frac{e^{o_{ij}}}{1 + e^{o_{ij}}}. \quad (11)$$

Equation (10) is the loss function of RankNet by cross entropy, where o_{ij} is the difference between the predicted score for document *i* and document *j*, namely $o_{ij} = f(x_i) - f(x_j)$. P_{ij} is the predicted probability of the document pair, and Q_{ij} is the target probablitity, which can be computed using the ground truth labels in a similar way as (11).

For listwise approach, the loss function is measured in terms of the difference between the target ranking list and the predicted ranking list of documents [26]. In essence, it is nearer to the concept of ranking. During the model training, it tunes the model to fit the predicted ranking for the target ranking until a balance between its prediction and the target.

ListNet [31] is a listwise learning to rank method, which can be taken as the listwise version of RankNet. Similar to RankNet, ListNet also takes the neural network as the basic ranking model, and employs probabilistic ranking loss function. Their difference lies in the definition of the loss function. Unlike RankNet's loss function defined upon document pairs, ListNet's loss function is defined on the probability of ranking lists, namely the permutation probability

$$\text{loss}(y, z(f_w)) = \sum_{j=1}^{n(i)} P_y(x_j)\log(P_{z(f_w)}(x_j)). \quad (12)$$

Equation (12) is the loss function of ListNet, which is based on the cross entropy of the target ranking list and the predicted ranking list. In the equation, *y* is the target ranking list for a given query, $z(f_w)$ is the predicted ranking list, where f_w is the scoring function with the weights *w*. $P_y(x_j)$ refers to the target ranking probability, and $P_{z(fw)}(x_j)$ is the predicted ranking probability.

LambdaMART [1], as a listwise method, is the boosted tree version of listwise LambdaRank [2], which is based on RankNet [27]. LambdaMART, as an ensemble of tree-based rankers, implements LambdaRank using Multiple Additive Regression Tree (MART) [28]. MART outputs its model as a linear combination of a set of regression trees. In its learning process, MART learns the next regression tree through performing gradient descent in function space, and outputs an ensemble of regression trees in its final model. LambdaMART uses MART with specific appropriate gradients and the Newton step to find the minimum of the loss function, and then compute output values of leaf nodes in each regression tree. Similar to MART, LambdaMART utilizes gradient boosting to optimize its loss function, which produces an ensemble of weak learners to form a strong one. In order to train a model, we don't need the costs themselves but the gradients. LambdaRank introduces parameter **λ** as a replacement of the loss function gradient. The **λ** for a given document in the ranking list gets contributions from all other documents under the same query with different labels

$$\lambda_i = \sum\nolimits_{j:(i,j)\in I} \lambda_{i,j} - \sum\nolimits_{j:(j,i)\in I} \lambda_{i,j}. \quad (13)$$

The loss function of LambdaMART has the same form as RankNet based on a probability function combining the score of each document. LambdaMART modifies the gradient of the loss with the variation of ranking performance through swapping the rank positions of the two documents as shown in (13), where λ_{ij} is the ranking loss by swapping the positions of the document *i* and the document *j*. LambdaMART uses **λ** as the gradient of loss function and uses boosted regression trees as its model to decrease ranking loss in iterations as MART does. Readers can refer to [29] for details about LambdaMART algorithm.

To help understand the proposed framework based on learning to rank methods for biomedical document retrieval, we illustrate the detailed steps of our framework in Fig. 1.

Overall, there are two phases to use learning to rank methods for document ranking, namely the training phase and the testing phase.

Before the training phase, we need to represent each document as a feature vector, which encodes the abundant information of the original document by taking different factors related to the document into consideration. As a common way in information retrieval, we extract these factors using different kinds of traditional retrieval models

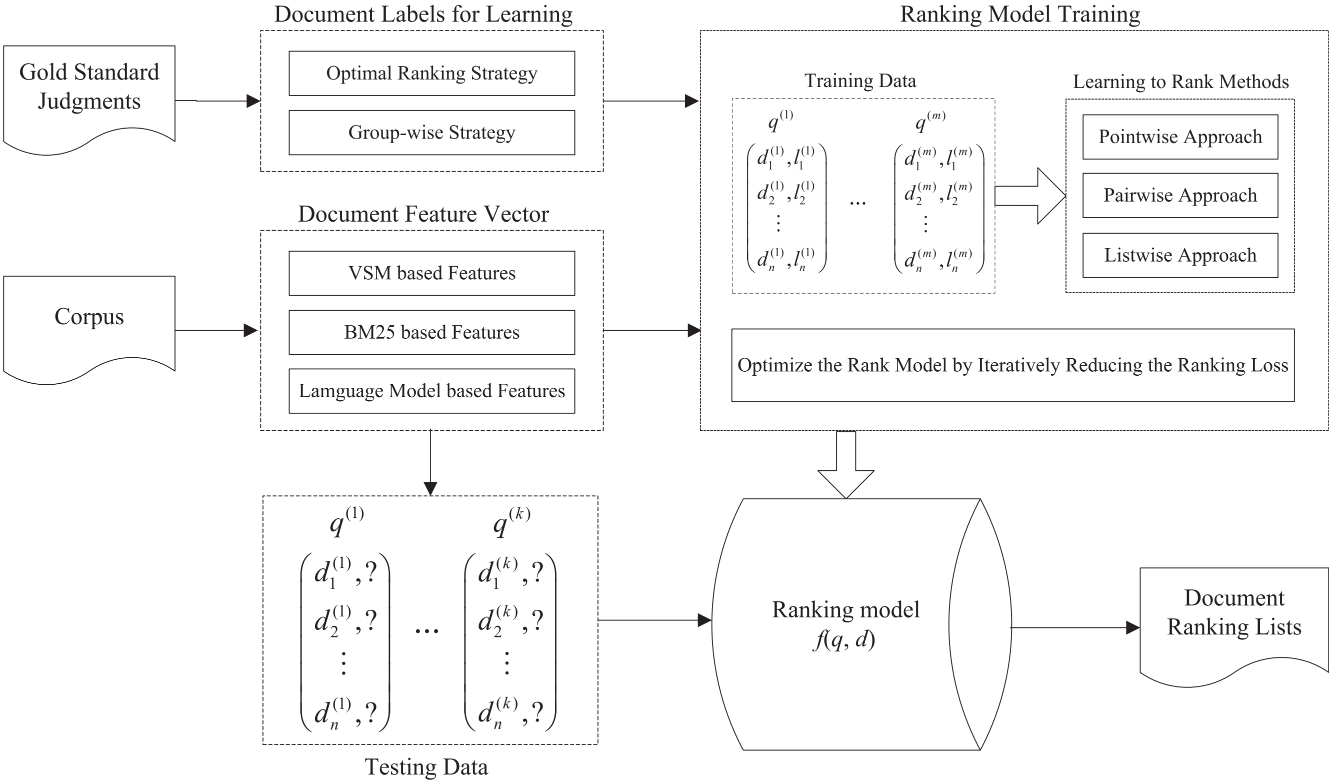


Fig. 1. The processing flow of the proposed framework based on learning to rank methods.

including the vector space model, BM25 model and the probablistic language models. As mentioned in the former sections, we switch the parameters of these models to obtain different features of the document.

Meanwhile, we need to give each document a label in consideration of both the relevance-based and the diversity-based ground truths. We take the obtained document labels as the learning target for constructing the learning to rank based ranking models. Specifically, we use the proposed two labeling strategies to obtain the labels, namely the optimal ranking labels and group-wise labels.

In the training phase, we take the feature vectors and the labels of the documents with respect to the training queries as the input. Learning to rank methods optimize the learned ranking model by iteratively reducing the ranking loss, outputting the final ranking models. We conputes the ranking loss using predefined loss functions, which measures the difference between the predicted ranking and the target ranking, and different learning to rank methods model the ranking loss in different ways.

In the testing phase, we used the learned ranking model to predict the ranking list of the documents related to the testing queries, where the documents in the testing data is also represented as feature vectors without the labels. The predicted document ranking lists are the final retrieval results of our framework, and we evaluate the performance of our framework mainly based on these retrieval results.

4 Experiments and Analysis

4.1 Experimental Settings

We examine our learning framework on TREC Genomics track 2006 & 2007 datasets. The dataset consists of 162,259 documents from 49 genomics-related journals. These documents are divided into more than 10 million passages based on the pre-defined passage legal spans, which is provided by TREC committee and used as the standard division for original documents [9]. There are totally 62 queries, 26 queries of which are from 2006's track (we remove two queries with no relevant documents in advance) and 36 queries are from 2007's track.

We perform five fold cross validation to examine the performance. Specifically, queries from 2006 and 2007 TREC Genomics tracks are respectively divided by the query number into training set, validation set and testing set, where 60 percent queries are used for training, 20 percent for validation and 20 percent for testing. The training set is utilized for training ranking models, the validation set is utilized for the model selection in terms of different parameters and the test set is utilized for predicting on new queries. The reporting results are averaged over all the folds. Because our framework is general, other biomedical collections can also be applied. We choose the TREC collection to facilitate the comparisons with others' work. The retrieval units for the datasets are passages, so we will replace the phrase "document retrieval" with the phrase "passage retrieval" in our experiments, but in practice, they are the same.

4.2 Evaluation Measures

We evaluate the retrieval performance of the proposed framework by taking the evaluation measures used in TREC Genomics Track, Document MAP, Aspect MAP, Passage MAP and Passage2 MAP [9] [10]. These variations of Mean Average Precision (MAP) can help measure both the diversity and the relevance of retrieved passages. In this section, we will introduce these measures in details.

Document MAP counts all the PMIDs that have a passage associated with a topic ID in the set of gold standard passages as a relevant document for that topic, and all other documents are considered not relevant for that topic. For a given submission run, average precision is measured at each point of relevant PMIDs for a topic, and counts only the first time, when more than one passages have the same PMID. The MAP is the mean of the average precisions across topics.

Passage MAP, as a variation of MAP, computes individual precision scores for passages based on character-level precision. Specifically, for each relevant retrieved passage, passage MAP counts the number of characters overlapped with those ground truth relevant passages, and computes as the number of the overlapped characters divided by the total number of characters in the corresponding ground truth relevant passage. Then the mean of these average precisions over all topics is calculated to compute the MAP for passages.

Passage2 MAP is utilized in the 2007's track as a modification of Passage Map because of one shortcoming of the original Passage Map that some non-content manipulations of passages may largely impact the values. In this measure, MAP is caluculated as if each character in each passage were a ranked document. In essence, the output of passages is concatenated, with each character being from a relevant passage or not.

Aspect MAP is measured using the average precision for the aspects of a topic, averaged across all topics. To compute this, aspect MAP transforms the ranked passages to a ranked list of aspects. MAP is calculated similar to how it is calculated for documents, with the additional wrinkle that a single passage may have associated with multiple aspects. Therefore the precision for the retrieval of each aspect was computed as the fraction of relevant passages for the retrieved passages up to the current passage under consideration. These fractions at each point of first aspect retrieval were then averaged together to compute the average aspect precision. Taking the mean over all topics produced the final aspect-based MAP.

These variations of MAP are incomparable directly because they measure the precision at recall of different things. Typically, aspect MAP cn measure the diversity degree or the completeness of the ranking list from the aspect level, and the other three metrics measures the relevance degree of the ranking list from document and character level. Overall, we take all the four measures to examine the retrieval performance comprehensively.

4.3 Topics and Gold Standard Judgments

In this section, we will give a detailed introduction to the topics and gold standard judgments for the genomics tracks, which would help better understand our experiments.

For the TREC genomics track, the task is to retrieve for answers to some pre-defined biomedical questions, whose ground truth answers have been judged in advance. An answer can be any passage containing pieces of continuous text within a document relevant to the question.

We give a sample of queries for 2006's task and queries for 2007's task respectively shown in Table 1. The only difference for the two year's track lies in that the answer types of 2007's topics are defined within the question itself. For example, one of the questions in the track is "What serum [PROTEINS] change expression in association with high disease activity in lupus?", where PROTEINS is the answer type. In our experiments, we take the answer type as one of the query terms.

TABLE 1
Sample Topics for 2006's and 2007's Tracks

ID	2006's Topics
160	What is the role of PrnP in mad cow disease?
161	What is the role of IDE in Alzheimer disease?
162	What is the role of MMS2 in cancer?
163	What is the role of APC adenomatous polyposis coli in colon cancer?
164	What is the role of Nurr-77 in Parkinson disease?
ID	2007's Topics
200	What serum [PROTEINS] change expression in association with high disease activity in lupus?
201	What [MUTATIONS] in the Raf gene are associated with cancer?
202	What [DRUGS] are associated with lysosomal abnormalities in the nervous system?
203	What [CELL OR TISSUE TYPES] express receptor binding sites for vasoactive intestinal peptide VIP on their cell surface?
204	What nervous system [CELL OR TISSUE TYPES] synthesize neurosteroids in the brain?

Gold standard judgments in the corpus have a major difference compared with the relevance judgments in general information retrieval tasks, where the judgment is usually based on the relevance degree. However, in the judgment for the genomics task, there are two steps. Firstly, human experts annotate the retrieved passages as relevant or not. Secondly, the annotated relevant passages are given some terms indicating some query-related aspects, which is important for measuring the diversity of search results. For example, a piece of the gold standard judgments can be '160 11278343 5039 508 NEUROLOGIC MANIFESTATIONS|PRIONS/CHEMISTRY|PRPSC PROTEINS', in which the first item 160 is the topic ID, and the triple '11278343 5039 508' is the passage identifier. In the triple, '11278343' is the PMID for the document, '5039' is the beginning character of the passage in the document, and '508' is the length of the passage. The last item is the query-related aspects for the passage, which are selected from Medical Subject Headings (MeSH) terms. The aspects are separated using a character '|', when the passage covers more than one aspects.

4.4 Ranking Model Compared

To compare the retrieval effectiveness of our proposed framework, we evaluate the following ranking models in our experiments; there are totally five kinds of ranking models.

(a) The ranking model obtained from original submission run. The original ranking score can be taken as the baseline model in our experiments. For 2007 queries, we select two official submission runs and an Okapi run. The first one is NLMinter developed

TABLE 2
Retrieval Performance of Ranking Models on the TREC 2006 Genomics Collection

MAP	Document	Passage	Aspect	Passage2
Okapi	0.3466	0.0282	0.2362	0.0325
Survival	0.3523	0.0290	0.2450	0.0331
LTR	0.3490	0.0291	0.2351	0.0312
Opi_Rank	0.3522	0.0300	0.2443*	0.0349*
	(+1.62%)	(+6.28%)	(+3.43%)	(+7.41%)
Group	0.3780*	0.0450*	0.2494*	0.0619*
	(+9.07%)	(+59.45%)	(+5.58%)	(+90.27%)

TABLE 3
Retrieval Performance of Ranking Models on the TREC 2007 Genomics Collection

MAP	Document	Passage	Aspect	Passage2
Okapi	0.2562	0.0659	0.1948	0.0800
Survival	0.2654	0.0720	0.2022	0.0853
LTR	0.2579	0.0707	0.1947	0.0812
Opi_Rank	0.2640	0.0715*	0.2138*	0.0821
	(+3.06%)	(+7.83%)	(+9.77%)	(+2.61%)
Group	0.3555*	0.1125*	0.2822*	0.1226*
	(+38.77%)	(+57.33%)	(+44.90%)	(+53.30%)
NLMinter	0.3286	0.0968	0.2631	0.1148
Survival	0.3243	0.0969	0.2695	0.1183
LTR	0.3270	0.0953	0.2644	0.1135
Opi_Rank	0.3309	0.0972*	0.2638*	0.1152
	(+0.69%)	(+0.37%)	(+0.27%)	(+0.35%)
Group	0.4264*	0.1211*	0.2896*	0.1425*
	(+29.75%)	(+25.07%)	(+10.09%)	(+24.12%)
MuMSHfd	0.2906	0.0840	0.2068	0.0895
Survival	0.2844	0.0844	0.2256	0.0918
LTR	0.2903	0.0791	0.2097	0.0860
Opi_Rank	0.2941	0.0975*	0.2152*	0.0988*
	(+1.22%)	(+16.13%)	(+4.04%)	(+10.34%)
Group	0.2991*	0.0977	0.2220*	0.1033*
	(+2.94%)	(+16.28%)	(+7.32%)	(+15.38%)

by the U.S. National Library of Medicine [12], which achieves the best performance in the 2007's track in terms of Aspect MAP, Passage2 MAP and Document MAP. The second one is MuMshFd, which is also one of the best performed runs in 2007's track [30]. The last one is solely based on the probabilistic weighting model BM25. Its performance is above average among all the results reported in the TREC 2007 Genomics track. For 2006 queries, we only select the Okapi run as our baseline, because other official submissions are not available. We conduct re-ranking experiments on the top 1,000 passages of each baseline run, which is the maximum size of a standard submission.

(b) The ranking model obtained using survival modeling approach in [14]. In this approach, survival analysis is introduced for modeling aspects to promote ranking diversity in biomedicine domain, which can be considered as a strong baseline.

(c) The ranking model obtained from traditional learning to rank methods. This is another class of baseline models based solely on traditional learning to rank models. These models contain all the features described in the paper, but only label passages as relevant or not. We denote these models as traditional LTR models in our study.

(d) The ranking model obtained by optimal labeling strategy. These ranking models utilize optimal labeling strategy to construct the ranking model with all the defined features.

(e) The ranking model obtained using group-wise labeling strategy. These ranking models utilize group-wise labeling strategy to construct the ranking model with all the defined features.

Besides, we compare three learning to rank approaches, namely the pointwise approach, the pairwise approach and the listwise approach in our study.

4.5 Comparisons on Retrieval Performance

In this section, we evaluate our methods based on the learning to rank method LambdaMART in comparison with all baseline runs, and show their performance in Tables 2 and 3, where Survival refers to the method in [14], LTR refers to the original learning to rank method, the Opi_Rank represents the method based on optimal ranking labeling strategy, and Group represents the methods based on group-wise learning to rank. The values in parentheses are the relative rates of improvement over the original results. Besides, we compare the results using statistical test (i.e., two-tailed paired Student's t tests), where '*' indicates that improvement of term ranking over original run is significant with 95 percent confidential level ($p < 0.05$).

From Table 2, we can see that when directly applying original learning to rank method, the retrieval performance is comparable with the original run, while the method based on survival analysis outperforms original Okapi run and LTR method. Learning framework based on optimal ranking strategy gains comparable results with Survival method, and outperforms other methods in all the evaluation measures. Group-wise learning to rank framework improves the retrieval performance further, and significantly outperforms the original run. The improving percentage on Passage MAP and Passage2 MAP are more than 50 percent, implying that our framework can effectively retrieve relevant passages, and rank relevant passages on the top of the rank list. The improvement on Aspect MAP implies that the re-ranking passages list is more diversified than the original one. Therefore, by learning to rank biomedical passages based on Okapi run of 2006 queries, we can obtain more relevant and diversified results.

We can observe a similar tendency in Table 3, where our learning framework based on optimal ranking outperforms the original run, and the group-wise learning to rank method improves the performance further. Compared with Survival method, Opi_Rank method gains comparable ranking results, while the Group method further improves the performance. It is worthwhile to observe that our group-wise framework improves the best performance official run about 25 percent on Passage MAP and Passage2 MAP, and 10 percent on Aspect MAP. That is to say, our framework, compared with the original run, obtains more relevant and diversified passages on the top to fulfill the information need.

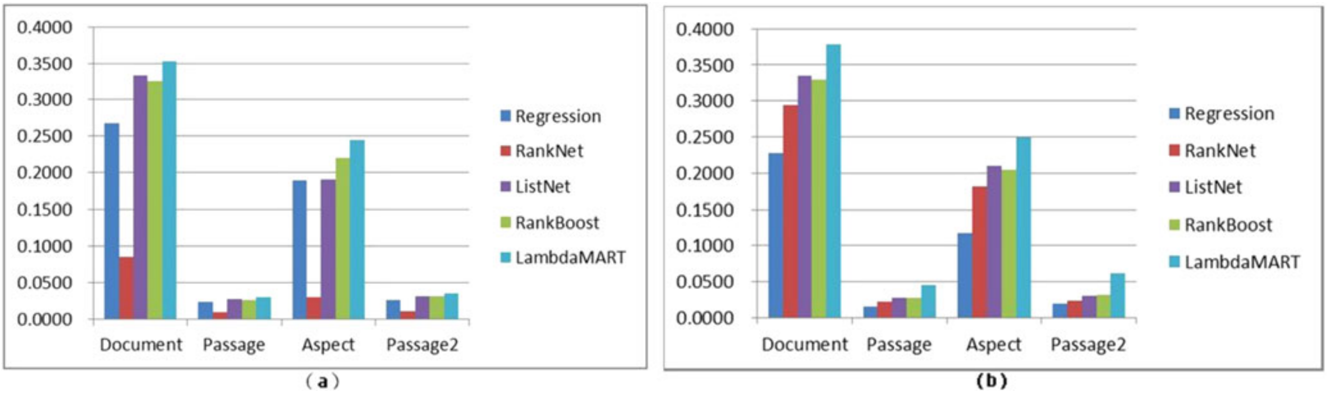


Fig. 2. Performance of different learning to rank methods based on the 2006's Okapi baseline. (a) The optimal ranking strategy. (b) The group-wise strategy.

From the tables, we can see that our methods achieve consistent improvement over all the baseline runs in terms of all levels of MAP evaluation measures. In comparison, the results based on optimal ranking labeling strategy outperform most of the baseline measures, and the group-wise learning to rank framework achieves better results, and improves the passage retrieval performance further.

4.6 Performance of Different Learning to Rank Approaches

In this section, we compare the effectiveness of our framework using five state-of-the-art learning to rank methods based on four baseline runs mentioned above. These methods belongs to three learning to rank approaches, which are Regression [25] (pointwise), RankNet [27] (pairwise) and RankBoost [4] (pairwise), ListNet [31] (listwise) and LambdaMART [1] (listwise). We respectively examine the retrieval performance of the methods based on the two proposed labeling strategies, optimal ranking strategy and the group-wise strategy. The performance is also measured in terms of document MAP, passage MAP, aspect MAP and passage2 MAP, respectively denoted as Document, Passage, Aspect and Passage2. The comparisons of results on these standard submission runs are shown in Figs. 2 to 5.

Fig. 2 presents the results based on Okapi run of Genomics track 2006's queries. From Fig. 2a, we can see that using the optimal ranking strategy, methods except RankNet gains comparative performance. Pairwise method RankBoost and listwise methods, ListNet and LambdaMART outperform pointwise methods Regression. The performance of RankBoost and ListNet tends to be on the same level, and LambdaMART performs the best among all the methods. We can also find that the similar tendency on Fig. 2b, except that the performance of RankNet based on group-wise strategy outperforms the pointwise Regression.

Fig. 3 presents the results based on Okapi run of Genomics track 2007's queries. From Fig. 3a, we can find a distinct tendency in terms of all the evaluation measures compared with the results in Fig. 2a, where the performance of RankBoost is higher than ListNet, but lower than LambdaMART, and RankNet still performs worse than all the other methods. In Fig. 3b, all the methods gain comparative performance on the same level, and LambdaMART performs the best.

Fig. 4 presents the results based on NLMinter baseline run of Genomics track 2007's queries, which is one of the best-performed submissions in the track. The tendency in Fig. 4a tends to be the same with that in Fig. 2a, and in Fig. 4b RankNet and ListNet gains the performance on the same level.

Fig. 5 presents the results based on MuMshFd baseline run of Genomics track 2007's queries, which is also one of the best performed runs in the track. The tendency in Fig. 5a tends to be the same as that in Figs. 2a and 3a, and the performances of all the methods in Fig. 5b are on the same level, difficult to tell which is better in terms of all the evaluation measures.

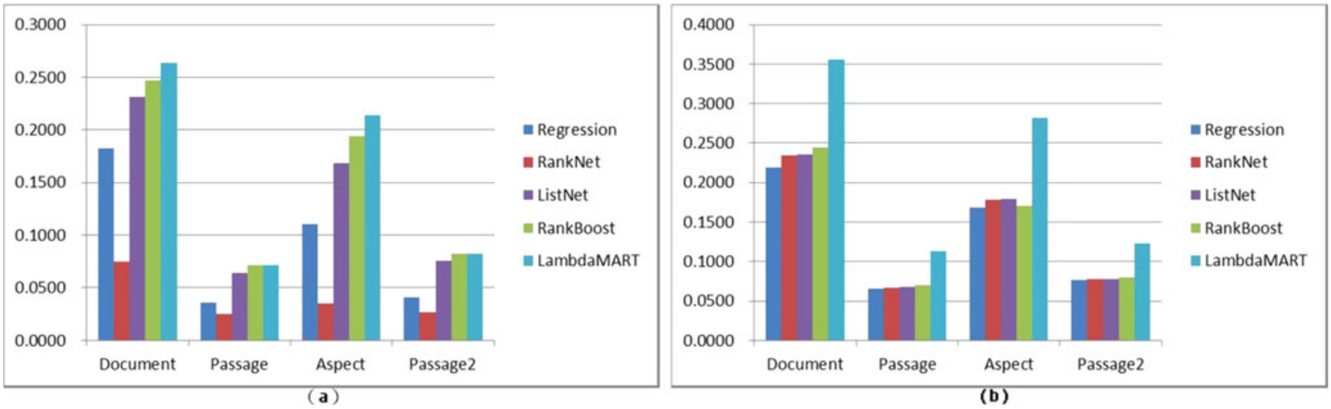


Fig. 3. Performance of different learning to rank methods based on the 2007's Okapi baseline. (a) The optimal ranking strategy. (b) The group-wise strategy.

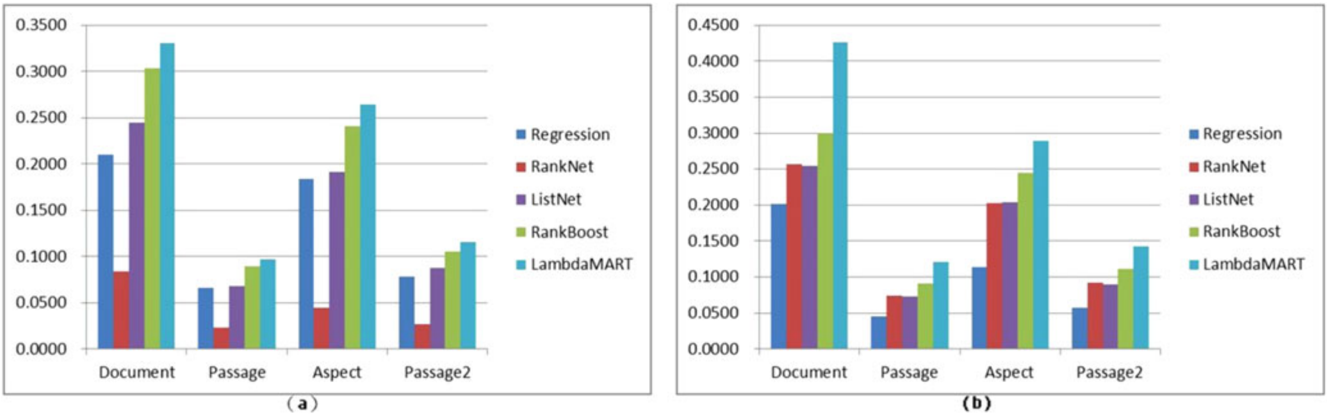


Fig. 4. Performance of different learning to rank methods based on the 2007's NLMinter baseline. (a) The optimal ranking strategy. (b) The group-wise strategy.

From these figures, we find that, compared with all the other methods, LambdaMART performs the best on the baseline runs in terms of most of the evaluation measures, while RankBoost performs a little better than ListNet under most of the experiments. Performance of Regression is slightly lower than the former three methods. The performance of RankNet varies a lot on the two strategies. For optimal ranking strategy, RankNet does not perform very well, but for the group-wise strategy, its performance is almost between Regression and other methods.

4.7 Discussion

In this section, we will further discuss and analyze our experimental results to find the advantages and disadvantages of our methods.

The optimal ranking labeling strategy can set a learning target for learning to rank algorithms to tune the model, and it seems effective to improve the original results. Meanwhile, the learning target may focus too much on the most diversified passages, so its performance is less significant on all the evaluation measures. In comparison, group-wise learning to rank can better meet the requirements for diversity-oriented retrieval by taking groups as a training unit, and each group consists of one or more diversified passage, some less diversified passages and a group of irrelevant passages. Based on this idea, learning to rank algorithm can be focused on the passages with more aspects, and tends to choose different aspects in various ways, resulting in more effective ranking models. Therefore, the ranking models can contribute more to the performance in terms of both relevance and diversity.

Besides, from the Algorithm 1 and Algorithm 2, we can also find that time complexity of group-wise learning to rank framework is much lower than the optimal ranking one. Above all, we believe that group-wise learning to rank framework is more effective than the optimal ranking framework for biomedical document retrieval to improve the performance in terms of relevance and diversity.

5 Conclusions and Future Work

In this paper, we propose a learning to rank based framework for biomedical information retrieval, focusing on improving the retrieval performance in term of both relevance and diversity. The proposed methods are respectively based on optimal ranking strategy and group-wise learning to rank, seeking to boost the diversity of retrieved relevant documents. Besides, we investigate the effectiveness of our methods on various learning to rank methods belonging to three approaches, pointwise approach, pairwise approach and listwise approach. Experimental results on TREC Genomics track datasets demonstrate our proposed framework is effective in improving the performance of biomedical

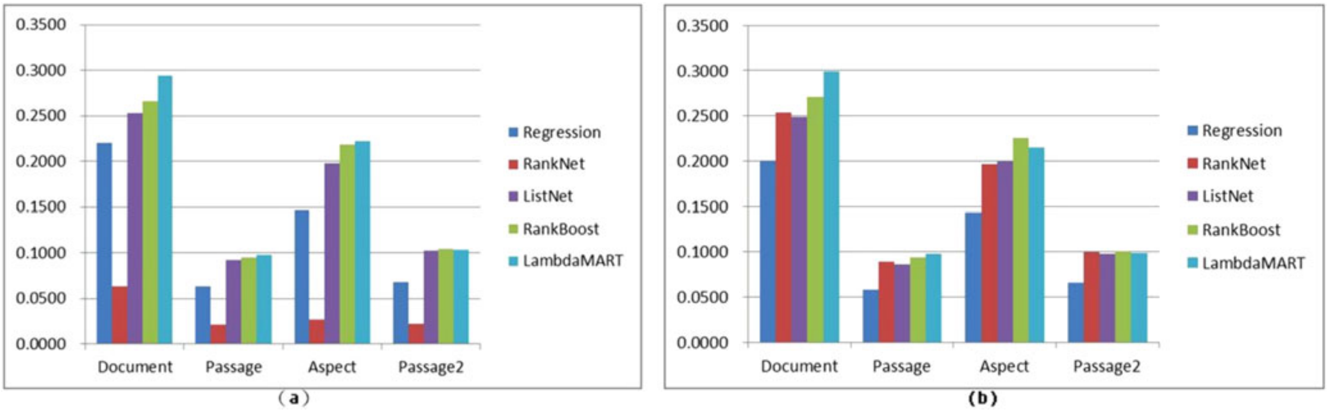


Fig. 5. Performance of different learning to rank methods based on the 2007's MuMshFd baseline. (a) The optimal ranking strategy. (b) The group-wise strategy.

retrieval. Learning to rank method, LambdaMART, outperforms other methods in our framework for biomedical retrieval. The optimal ranking strategy and the group-wise strategy can both contribute to the performance, and group-wise learning to rank can improve the performance better.

We will extend our future work in some directions. Since our proposed method needs explicit aspect annotations to train a ranking model, we will attempt to explore an approach for automatic aspect mining when the dataset contains no such annotations. Besides, we will attempt to incorporate an aspect-related item into the loss function for learning to rank methods, which may produce a more effective model. We will also develop and examine the performance of other features, especially some domain specific features, to make the framework more applicable for biomedical document retrieval.

Acknowledgments

This work is partially supported by grants from the Natural Science Foundation of China (Nos. 61272370, 61402075, 61572102, 61572098, 61272373), Natural Science Foundation of Liaoning Province, China (Nos. 201202031 and 2014020003), State Education Ministry and The Research Fund for the Doctoral Program of Higher Education (No. 20090041110002), and the Fundamental Research Funds for the Central Universities.

References

[1] C.J. Burges, "From ranknet to lambdarank to lambdamart: An overview," *Learning*, vol. 11, pp. 23–581, 2010.

[2] C.J. Burges, R. Ragno, and Q.V. Le, "Learning to rank with non-smooth cost functions," in *Proc. Neural Inf. Process. Syst.*, 2006, pp. 193–200.

[3] Y. Cao, J. Xu, T.Y. Liu, H. Li, Y. Huang, and H.W. Hon, "Adapting ranking SVM to document retrieval," In *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2006, pp. 186–193.

[4] 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.

[5] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. 8th Int. Conf. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2002, pp. 133–142.

[6] T.Y. Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retr.*, vol. 3, no. 3, pp. 225–331, 2009.

[7] Q. Wu, C.J. Burges, K.M. Svore, and J. Gao, "Ranking, boosting, and model adaptation," Microsoft Research, Cambridge, United Kingdom, Technical Report, MSR-TR-2008–109, 2008.

[8] J. Xu and H. Li, "Adarank: A boosting algorithm for information retrieval," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2007, pp. 391–398.

[9] W.R. Hersh, A.M. Cohen, P.M. Roberts, and H.K. Rekapalli, "TREC 2006 genomics track overview," in *Proc. Text REtrieval Conf.*, 2006.

[10] W.R. Hersh and E. Voorhees, "TREC genomics special issue overview," *Inf. Retr.*, vol. 12, no. 1, pp. 1–15, 2009.

[11] A. Goldbery, D.A.J. Gael, B. Settles, X. Zhu, and M. Craven, "Ranking biomedical passages for relevance and diversity," University of Wisconsin, Madison at TREC Genomics 2006; in *Proc 15th Text REtrieval Conf.*, 2006.

[12] D. Demner-Fushman, et al., "Combining resources to find answers to biomedical questions". in *Proc. Text REtrieval Conf.*, 2007.

[13] X. Yin, X. Huang, and Z. Li, "Promoting ranking diversity for biomedical information retrieval using wikipedia," in *Advances in Information Retrieval*, Berlin Heidelberg, Germany: Springer, 2010, pp. 495–507.

[14] X. Yin, J.X. Huang, Z. Li, and X. Zhou, "A survival modeling approach to biomedical search result diversification using wikipedia," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1201–1212, Apr. 2013.

[15] X. Yin, Z. Li, J.X. Huang, and X. Hu, "A relevance-novelty combined model for genomics search result diversification," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2010, pp. 692–695.

[16] Y. Chen, X. Yin, Z. Li, X. Hu, and J.X. Huang, "Promoting ranking diversity for biomedical information retrieval based on LDA," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2011, pp. 456–461.

[17] X. An, and J.X. Huang, "Boosting novelty for biomedical information retrieval through probabilistic latent semantic analysis," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2013, pp. 829–832.

[18] T. Qin, T.Y. Liu, J. Xu, and H. Li, "LETOR: A benchmark collection for research on learning to rank for information retrieval," *Inf. Retr.*, vol. 13, no. 4, pp. 346–374, 2010.

[19] 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.

[20] 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.

[21] Y. Lin, H. Lin, Z. Ye, S. Jin, and X. Sun, "Learning to rank with groups," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1589–1592.

[22] R. Baeza-Yates and B. Ribeiro-Neto, "Modern information retrieval," vol. 463. New York, NY, USA: ACM Press, 1999.

[23] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3," Gaithersburg, MD, USA: NIST, 1995, pp. 109–109.

[24] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2001, pp. 334–342.

[25] D. Cossock and T. Zhang, "Subset ranking using regression," in *Learning theory*. Berlin Heidelberg, Germany: Springer, 2006, pp. 605–619.

[26] F. Xia, T.Y. Liu, J. Wang, W. Zhang, and H. Li, "Listwise approach to learning to rank: Theory and algorithm," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1192–1199.

[27] C.J. Burgeset al., "Learning to rank using gradient descent," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 89–96.

[28] J.H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[29] Y. Ganjisaffar, R. Caruana, and C.V. Lopes, "Bagging gradient-boosted trees for high precision, low variance ranking models," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2011, pp. 85–94.

[30] N. Stokes, Y. Li, L. Cavedon, E. Huang, J. Rong, and J. Zobel, "Entity-based relevance feedback for genomic list answer retrieval," presented at the Text REtrieval Conf., Gaithersburg, MD, USA, 2007.

[31] 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.

[32] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *Proc. 2nd ACM Int. Conf. Web Search Data Min.*, 2009, pp. 5–14.

[33] F. Radlinski, R. Kleinberg, and T. Joachims, "Learning diverse rankings with multi-armed bandits," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 784–791.

[34] C.X. Zhai, W.W. Cohen, and J. Lafferty, "Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2003, pp. 10–17.

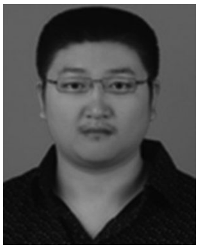
[35] B. Carterette and P. Chandar, "Probabilistic models of ranking novel documents for faceted topic retrieval," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1287–1296.

[36] S. Vargas, P. Castells, and D. Vallet, "Explicit relevance models in intent-oriented information retrieval diversification," in *Proc. 35th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, 2012, pp. 75–84.

[37] Y. Yue and T. Joachims, "Predicting diverse subsets using structural SVMs," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1224–1231.

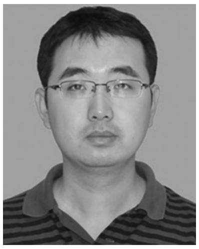
Bo Xu received the BSc degree from the Dalian University of Technology, China, in 2011. He is currently working toward the PhD degree from the School of Computer Science and Technology, Dalian University of Technology. His current research interests include information retrieval and learning to rank.

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 at the 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.

Yunlong Ma received the BSc degree from the Dalian University of Technology, China, in 2009. He is currently working toward the PhD degree from the School of Computer Science and Technology, Dalian University of Technology. His current research interests include information retrieval and relevance feedback.



Liang Yang received the BSc degree from the Dalian University of Technology, China, in 2009. He is currently working toward the PhD degree from the School of Computer Science and Technology, Dalian University of Technology. His current research interests include information retrieval and sentiment analysis.

Jian Wang received the BSc, MSc, and PhD degrees from the Dalian University of Technology, China, in 1988, 1993, and 2014, respectively. She is currently a professor in the School of Computer Science and Technology, Dalian University of Technology. Her current research interests include biomedical literature data mining, information retrieval, and natural language processing.

Zhihao Yang received the PhD degree in computer science from the Dalian University of Technology, China, in 2008. He is currently a professor in the School of Computer Science and Technology, Dalian University of Technology. He has published more than 20 research papers on topics in biomedical literature data mining. His current research interests include biomedical literature data mining and information retrieval.