Global Ranking of Documents Using Continuous Conditional Random Fields

[1]Tao Qin, [1]Tie-Yan Liu, [2]Xu-Dong Zhang, [2]De-ShengWang, [1]Hang Li
[1]Microsoft Research Asia, [2]Tsinghua University
[1]{taoqin,tyliu,hangli}@microsoft.com
[2]{zhangxd, wangdsh_eeg}@tsinghua.edu.cn

May, 2008

Abstract

This paper is concerned with ranking model construction in document retrieval. Traditionally, the ranking model is defined as a function of a query and a document. In practice, many factors affecting ranking can and must be taken into consideration, for instance, similarities between documents and hyper links between documents. One needs to exploit a new ranking model which is a function of a query and the entire set of documents retrieved with the query. This paper names this new problem 'global ranking of documents', in contrast to traditional 'local ranking of documents'. The paper proposes a novel learning to rank method to perform the task. The method employs Continuous Conditional Random Fields (CRF) as model, which is a conditional probability distribution representing the mapping relationship from the retrieved documents to their ranking scores. The model can naturally utilize as features the content information of documents as well as the relation information between documents for global ranking. A learning algorithm for creating Continuous CRF is also presented in the paper. Taking Pseudo Relevance Feedback and Topic Distillation as examples, this paper shows how the learning method can be applied to global ranking. Experimental results on benchmark data show that the proposed method outperforms the baseline methods.

1 Introduction

Ranking is a central issue for search, because the goodness of a search system is mainly evaluated by the accuracy of its ranking results. Traditionally, the ranking model is defined as a function of a query and a document, which represents the relevance of the document with respect to the query. In search, given a query the ranking model assigns a score to each of the documents retrieved with the query and outputs the ranked list of documents sorted by the scores. The ranking model is a local model in the sense that the creation and utilization of the model only needs the involvement of a single document. In this paper, we refer to this problem setting 'local ranking of documents'.

As search evolves, more and more useful information for ranking becomes available. This includes the content information of documents as well as the relation information between documents. The relations can be hyper links between documents, similarities between documents, etc. Ideally, the ranking model would be a function of the query and all the retrieved documents with the query. That is to say, ranking should be conducted on the basis of the contents of documents as well as the relations between documents. We refer to this setting 'global ranking of documents'. Obviously global ranking contains local ranking as its special case.

The necessity and importance of conducting global ranking has already been recognized in IR. However, it is only considered as separated issues. For instance, application tasks such as Topic Distillation, Pseudo Relevance Feedback, and Subtopic Retrieval were studied.

In this paper we investigate how to conduct global ranking with a *general and principled approach*. (1) We first give a formal definition of the problem of global ranking. (2) We then propose employing a Continuous CRF model for conducting global ranking. The Continuous CRF model is defined as a conditional probability distribution representing the mapping relationship from the retrieved documents to their ranking scores, where the ranking scores are represented by *continuous* variables. In Continuous CRF, we make use of both content information and relation information as features. The model is general in the sense that various types of relations for global ranking can be incorporated as features. (3) We further propose a learning method for training Continuous CRF. Specifically, we use Maximum Likelihood Estimation and Gradient Ascent for parameter estimation. Our learning method turns out to be a novel learning-to-rank method for global ranking, in contrast to the exiting learning-to-rank methods which are by design local ranking methods. (4) We apply Continuous CRF to Pseudo Relevance Feedback and Topic Distillation. Experimental results on benchmark data show that our method performs better than the baseline methods.

The remaining part of the paper is organized as follows. In Section 2, we introduce related work. We give a formal definition of global ranking in Section 3. We define Continuous CRF for global ranking in Section 4. We then show how to use Continuous CRF in Pseudo Relevance Feedback and Topic Distillation in Section 5. We give experimental results in Section 6. Finally, we conclude the paper in the last section.

2 Related work

2.1 Ranking Using Relation Information

Traditionally, document ranking was only conducted locally, in the sense that the ranking model is a function of a query and a single document. Although this makes the ranking model easy to create and use, its limitation is also clear. There is a large amount of information which is useful for ranking, but is not local, for example, the relation information between documents.

Relation information between documents plays an important role in many information retrieval tasks. For example, ranking web pages on the basis of importance, improving relevance ranking by using similarity information, and diversifying search

results.

Relation information has been used for importance ranking in web search. PageRank[21] and HITS[14] are well known algorithms for computing importance of web pages. They rank web pages based on the Markov chain model and authority-hub model respectively; both leverage the hyperlink (relation) information between web pages.

Topic Distillation [32, 25] is another example of using relation information in web search. Here, Topic Distillation refers to the search task in which one selects a page that can best represent the topic of the query from a web site by using structure (relation) information of the site. If both a page and its parent page are concerned with the topic, then the parent page is to be ranked higher. It is found that propagating the relevance of a web page to its neighborhood through the hyperlink graph can improve the accuracy of Topic Distillation [27]. Furthermore, propagating the relevance of a web page to its parent page can also boost the accuracy [24].

Similarity between documents is useful information for relevance search. In Pseudo Relevance Feedback [5, 6, 7, 30, 15], we first conduct a round of relevance ranking, assuming that the top ranked documents are relevant; then conduct another round of ranking, using similarity information between the top ranked documents and the other documents, and boost some relevant documents dropped in the first round. Existing Pseudo Relevance Feedback methods can be clustered into two groups. In one group, the documents are ranked first based on relevance. Then, the top results are used to make an expansion of the query and the re-ranking with the expanded query is performed [30]. In the other group [15, 28], it is assumed that *similar documents are likely to have similar ranking scores*, and documents are re-ranked based on the similarities between documents.

In Subtopic Retrieval one also utilizes document similarity information [34]. In the task, given a query, the returned documents should cover as many subtopics as possible. If there are multiple documents about the same subtopic, then only one document should be selected and ranked high.

Although relation information between documents has been used in search, so far there has been no previous work which generalizes the specific ranking tasks.

2.2 Learning to Rank

Recently machine learning technologies called 'learning to rank' have been applied to information retrieval. In the approach, supervised learning is employed in ranking model construction. Previous work demonstrates that learning to rank has certain advantages when compared with the traditional non-learning approaches.

Previously people have tried to transform the problem of ranking into that of classification and apply existing classification techniques to the task. This is called the pairwise approach in the literature. For example, as classification techniques one can employ SVM and Neural Network, and derive the methods of RankSVM [10, 13] and RankNet [2]. See also [31, 35, 23]. More recently, a number of authors have proposed directly defining a loss function on list of objects and optimizing the loss function in learning [3, 33]. This listwise approach formalizes the ranking problem in a more straightforward way and thus appears to be more effective.

All the learning to rank methods, however, do not consider using the relation information between objects (documents) in the models. As a result, they are not directly applicable to the cases in which relation information should be used. Making extensions of existing methods on the basis of heuristics would not work well, as will be seen in the experiment section.

2.3 Conditional Random Fields

Conditional Random Fields (CRF) is a discriminative model for sequence data prediction. It is defined as a conditional probability distribution of output label sequence given input observation sequence [29]. The conditional probability distribution is an exponential model containing features based on both the input and output. It is assumed that there exists dependency between the adjacent labels in the output.

CRF was first applied to sequential data labeling [16] such as shallow parsing [26], named entity recognition [20], and table extraction [22]. Later, it was also applied to other problems such as web information extraction [36].

Sutton and McCallum give a tutorial on CRF [29]. They conclude in the paper: "Conditional Random Fields are a natural choice for many relational problems because they allow both graphically representing dependencies between entities, and including rich observed features of entities." CRF is powerful because it is easy to include independent and non-independent features in the model [19], where the non-independent features can represent various types of relations.

In this paper, we apply CRF to document ranking. As far as we know, this is the first time that CRF is used in such kind of applications.

3 Global Ranking Problem

Document ranking in search is a problem as follows. When the user submits a query, the search system retrieves all the documents containing the query, calculates a ranking score for each of the documents using the ranking model, and sorts the documents according to the ranking scores. The scores determine the ranking orders of documents, and can indicate relevance, importance, and/or diversity of documents.

Let q denote a query. Let $d^{(q)} = \{d_1^{(q)}, d_2^{(q)}, \ldots, d_{n_{(q)}}^{(q)}\}$ denote the documents retrieved with q, and $y^{(q)} = \{y_1^{(q)}, y_2^{(q)}, \ldots, y_{n_{(q)}}^{(q)}\}$ denote the ranking scores assigned to the documents. Here $n_{(q)}$ stands for number of documents retrieved with q. Note that the numbers of documents vary according to queries. We assume that $y^{(q)}$ is determined by a ranking model.

We call the ranking 'local ranking', if the ranking model is defined as

$$y_i^{(q)} = f(q, d_i^{(q)}), i = 1, \ldots, n_{(q)} \tag{1}$$

Furthermore, we call the ranking 'global ranking', if the ranking model is defined as

$$y^{(q)} = F(q, d^{(q)}) \tag{2}$$

The major difference between the two is that F takes on all the documents as input, while f takes on individual documents as input. Note global ranking contains local

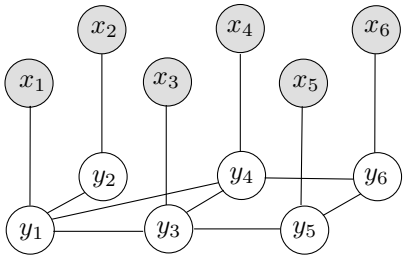

Figure 1: CRF Model

ranking as its special case. Intuitively, in local ranking we look at the documents individually, while in global ranking we treat the documents as a whole. In global ranking, we use not only the content information of documents but also the relation information between documents. There are many specific application tasks that can be viewed as examples of global ranking. These include Pseudo Relevance Feedback, Topic Distillation, and Subtopic Retrieval.

4 Ranking with Continuous CRF

We propose a learning to rank method for global ranking, using Continuous CRF as model.

4.1 Continuous CRF Model

Let $x = \{x_1, x_2, \ldots, x_n\}$ denotes the input vectors of the documents retrieved with the query, $y = \{y_1, y_2, \ldots, y_n\}$ denotes the ranking scores assigned to the documents, and R denotes the relation between the documents (and also the ranking scores of the documents). Here n stands for number of documents retrieved by the query and $x_i \in \mathscr{R}^{K1}, y_i \in \mathscr{R}, i = 1, 2, ..., n$ and $K1$ is an integer. Note that in this paper we call x input vector, not feature vector, in order to distinguish it from the feature functions in the CRF model. Let $\theta = \{\alpha, \beta\}, \alpha \in \mathscr{R}^{K1}, \beta \in \mathscr{R}^{K2}$ be a vector of parameters respectively, where $K2$ is an integer. Let $\{f_k(y_i, x)\}_{k=1}^{K1}$ be a set of real-valued feature functions defined on x and y_i ($i = 1, \ldots, n$), and $\{g_k(y_i, y_j, x)\}_{k=1}^{K2}$ be a set of real-valued feature functions defined on y_i, y_j, and x ($i = 1, \ldots, n$, $j = 1, \ldots, n$, $i \neq j$).

Continuous Conditional Random Fields (CRF) is a conditional probability distribution with density function:

$$\Pr(y|x) = \frac{1}{Z(x)} \exp\left\{\sum_i \sum_{k=1}^{K1} \alpha_k f_k(y_i, x) + \sum_{i,j} \sum_{k=1}^{K2} \beta_k g_k(y_i, y_j, x)\right\}, \tag{3}$$

where $Z(x)$ is a normalization function

$$Z(x) = \int_y \exp\left\{\sum_i \sum_{k=1}^{K1} \alpha_k f_k(y_i, x) + \sum_{i,j} \sum_{k=1}^{K2} \beta_k g_k(y_i, y_j, x)\right\} dy. \tag{4}$$

Continuous CRF is a graphical model, as depicted in Figure 1. In the conditioned undirected graph, a white vertex represents a ranking score, a gray vertex represents an input vector, an edge between two white vertexes represents relation between ranking scores, and an edge between a gray vertex and a white vertex represents dependency of ranking score on input vector. (In principle a ranking score can depend on all the input vectors; here for ease of presentation we only consider the simplified case in which it depends only on the corresponding input vector.)

In Continuous CRF, feature function f_k represents the dependency between the ranking score of a document and the input vector of the document, and feature function g_k represents the relationship between the ranking scores of two documents (e.g. similarity relation, parent-child relation). We call the feature functions f_k vertex features, and the feature functions g_k edge features. The edge feature functions are determined by the relation R. Different retrieval tasks have different definitions on R, as will be explained in Section 5.

There are clear differences between the conventional CRF and the Continuous CRF proposed here. (1) The conventional CRF is usually defined on a chain while Continuous CRF is defined on a graph. (2) In the conventional CRF random variable y is discrete while in Continuous CRF y is continuous. This makes the inference of Continuous CRF is very different from that of conventional CRF, as will be seen in Section 5. (3) The R in Continuous CRF defines the relations between ranking scores as well as specifies the corresponding feature function g_k. An R is associated with a query and different queries may have different R's.

4.2 Learning and Inference

In learning, given training data we estimate the parameters $\theta = \{\alpha, \beta\}$ of Continuous CRF. Suppose that the training data $\{x^{(q)}, y^{(q)}\}_{q=1}^{N}$ is generated i.i.d. from a unknown probability distribution, where each $x^{(q)} = \{x_1^{(q)}, x_2^{(q)}, ..., x_{n(q)}^{(q)}\}$ is a set of input vectors associated with the documents of query q, and each $y^{(q)} = \{y_1^{(q)}, y_2^{(q)}, ..., y_{n(q)}^{(q)}\}$ is a set of ranking scores associated with the documents of query q. Note that we do not assume that the documents of a query are generated i.i.d. Further suppose that the relation data $\{R^{(q)}\}_{q=1}^{N}$ is given, where each $R^{(q)}$ is a set of relations on the documents (ranking scores) of query q.

We employ Maximum Likelihood Estimation to estimate the parameters. Specifically, we calculate the conditional log likelihood of the data with respect to the Continuous CRF model.

$$L(\theta) = \sum_{q=1}^{N} \log \Pr(y^{(q)}|x^{(q)}; \theta). \tag{5}$$

Specifically,

$$L(\theta) = \sum_{q=1}^{N} \left\{ \sum_{i} \sum_{k=1}^{K1} \alpha_k f_k(y_i^{(q)}, x^{(q)}) + \sum_{i,j} \sum_{k=1}^{K2} \beta_k g_k(y_i^{(q)}, y_j^{(q)}, x^{(q)}) \right\} - \sum_{q=1}^{N} \log Z(x^{(q)}). \tag{6}$$

We take the parameter $\hat{\theta}$ that can maximize the log likelihood function as output. We know of no analytic solution to the optimization problem. We can employ the numerical method of Gradient Ascent to solve it.

In inference, given new data x and relation R we use the model with estimated parameter $\hat{\theta}$ to calculate the conditional probability $\Pr(y|x)$ and select the y satisfying

$$\begin{aligned} y &= \arg\max_y \Pr(y|x;\hat{\theta}) \\ &= \arg\max_y \left\{ \sum_i \sum_{k=1}^{K1} \hat{\alpha}_k f_k(y_i, x) + \sum_{i,j} \sum_{k=1}^{K2} \hat{\beta}_k g_k(y_i, y_j, x) \right\}. \end{aligned} \tag{7}$$

Equation (7) then becomes *the ranking model*.

5 Tasks

In this section we show how to apply Continuous CRF to two global ranking tasks.

5.1 Pseudo Relevance Feedback

Here we consider a method of using Continuous CRF for Pseudo Relevance Feedback, which belongs to the second group of existing methods (cf., Section 2.1). In fact, the method combines the two rounds of ranking of Pseudo Relevance Feedback into one by using the Continuous CRF model. In the method the similarities between any two documents are given and the input vectors are also defined. What we need to do is to specify the feature functions.

For each component in the input vector, we introduce a vertex feature function. Suppose that $x_{i,k}$ is the k-th component of input vector x_i, we define the k-th feature function $f_k(y_i, x)$ as

$$f_k(y_i, x) = -(y_i - x_{i,k})^2. \tag{8}$$

Next, we introduce one (and only one) edge feature function.

$$g(y_i, y_j, x) = -\frac{1}{2} S_{i,j} (y_i - y_j)^2, \tag{9}$$

where $S_{i,j}$ is similarity between documents d_i and d_j. The larger $S_{i,j}$ is, the more similar the two documents are. Here the relation R is represented by a matrix S, whose element in i-th row and j-th column is $S_{i,j}$. Note that this specific feature function does not depend on x.

The Continuous CRF for Pseudo Relevance Feedback then becomes

$$\Pr(y|x) = \frac{1}{Z(x)} \exp\left\{ \sum_i \sum_{k=1}^{K1} -\alpha_k (y_i - x_{i,k})^2 + \sum_{i,j} -\frac{\beta}{2} S_{i,j} (y_i - y_j)^2 \right\}, \tag{10}$$

where $Z(x)$ is defined as

$$Z(x) = \int_y \exp\left\{ \sum_i \sum_{k=1}^{K1} -\alpha_k (y_i - x_{i,k})^2 + \sum_{i,j} -\frac{\beta}{2} S_{i,j} (y_i - y_j)^2 \right\} dy. \tag{11}$$

To guarantee that $\exp\left\{\sum_i \sum_{k=1}^{K1} -\alpha_k(y_i - x_{i,k})^2 + \sum_{i,j} -\frac{\beta}{2} S_{i,j}(y_i - y_j)^2\right\}$ is integrable, we require that $\alpha_k > 0$ and $\beta > 0$.

The Continuous CRF can naturally model Pseudo Relevance Feedback. First, if the value of $x_{i,k}$ is high, then the value of y_i is high with high probability. (For example, $x_{i,k}$ can be a feature representing tf-idf.) Second, if the value of $S_{i,j}$ is large, then ranking scores y_i and y_j are close with high probability.

With some derivation, we obtain

$$Z(x) = (2\pi)^{\frac{n}{2}} |2A|^{-\frac{1}{2}} \exp(b^T A^{-1} b - c), \tag{12}$$

where $A = \alpha^T eI + \beta D - \beta S$, D is an $n \times n$ diagonal matrix with $D_{i,i} = \sum_j S_{i,j}$, I is an $n \times n$ matrix, $|A|$ is determinant of matrix A, $b = X\alpha$, $c = \sum_i \sum_{k=1}^{K1} \alpha_k x_{i,k}^2$, and X is a matrix whose i-th row k-th column element is $x_{i,k}$.

In learning, we try to maximize the log likelihood. Note that maximization of $L(\theta)$ in Eq. (6) is a constrained optimization problem because we need to guarantee that $\alpha_k > 0$ and $\beta > 0$. Gradient Ascent cannot be directly applied to a constrained optimization problem. Here we adopt a technique similar to that in [4] and then employ Gradient Ascent. Specifically, we maximize $L(\theta)$ with respect to $\log \alpha_k$ and $\log \beta$ instead of α_k and β. As a result, the new optimization issue becomes unconstrained.

The gradients of $L(\theta)$ with respect to $\log \alpha_k$ and $\log \beta$ are computed as follows.

$$\nabla_{\log \alpha_k} = \frac{\partial L(\theta)}{\partial \log \alpha_k} = -\alpha_k \sum_{q=1}^{N} \left\{ \sum_i (y_i^{(q)} - x_{i,k}^{(q)})^2 + \frac{\partial \log Z(x^{(q)})}{\partial \alpha_k} \right\} \tag{13}$$

$$\nabla_{\log \beta} = \frac{\partial L(\theta)}{\partial \log \beta} = -\beta \sum_{q=1}^{N} \left\{ \sum_{i,j} \frac{1}{2} S_{i,j}^{(q)} (y_i^{(q)} - y_j^{(q)})^2 + \frac{\partial \log Z(x^{(q)})}{\partial \beta} \right\} \tag{14}$$

Now we show how to get the partial derivative $\frac{\partial \log Z(x^{(q)})}{\partial \alpha_k}$ and $\frac{\partial \log Z(x^{(q)})}{\partial \beta}$. For simplicity, we omit the super script (q) in the equations hereafter. First

$$\frac{\partial \log Z(x)}{\partial \alpha_k} = -\frac{1}{2|A|} \frac{\partial |A|}{\partial \alpha_k} + \frac{\partial b^T A^{-1} b}{\partial \alpha_k} - \frac{\partial c}{\partial \alpha_k} \tag{15}$$

$$\frac{\partial \log Z(x)}{\partial \beta} = -\frac{1}{2|A|} \frac{\partial |A|}{\partial \beta} + \frac{\partial b^T A^{-1} b}{\partial \beta} \tag{16}$$

Furthermore,

$$\frac{\partial |A|}{\partial \alpha_k} = |A|(A^{-T}) :^T \frac{\partial A}{\partial \alpha_k} := |A|(A^{-T}) :^T I : \tag{17}$$

$$\frac{\partial |A|}{\partial \beta} = |A|(A^{-T}) :^T \frac{\partial A}{\partial \beta} := |A|(A^{-T}) :^T (D - S) : \tag{18}$$

$$\frac{\partial b^T A^{-1} b}{\partial \alpha_k} = X_{,k}^T A^{-1} b - b^T A^{-1} A^{-1} b + b^T A^{-1} X_{,k} \tag{19}$$

$$\frac{\partial b^T A^{-1} b}{\partial \beta} = -b^T A^{-1} (D - S) A^{-1} b, \tag{20}$$

Algorithm 1 Learning Algorithm of Continuous CRF for Pseudo Relevance Feedback

Input:training data $\{(x^{(1)}, y^{(1)}, S^{(1)}), (x^{(2)}, y^{(2)}, S^{(2)}), ..., (x^{(N)}, y^{(N)}, S^{(N)})\}$
Parameter: number of iterations T and learning rate η
Initialize parameter $\log \alpha_k$ and $\log \beta$
for $t = 1$ **to** T **do**
  for $i = 1$ **to** N **do**
    Compute gradient $\nabla_{\log \alpha_k}$ and $\nabla_{\log \beta}$ using Eq. (13) and (14) for a single query $(x^{(i)}, y^{(i)}, S^{(i)})$.
    Update $\log \alpha_k = \log \alpha_k + \eta \times \nabla_{\log \alpha_k}$ and $\log \beta = \log \beta + \eta \times \nabla_{\log \beta}$
  end for
end for
Output parameters of CRF model α_k and β.

where X : denotes the long column vector formed by concatenating the columns[1] of matrix X, and $X_{,k}$ denotes the k-th column of matrix X.

Substituting Eq.(15)-(20) into Eq. (13) and (14), we obtain the gradient of the log likelihood function. Algorithm 1 shows the learning algorithm based on Stochastic Gradient Ascent [2].

In inference, we calculate the ranking scores of documents with respect to a new query in the following way.

$$\hat{y} = \arg\max_{y} \Pr(y|x;\theta) = (\alpha^T eI + \beta D - \beta S)^{-1} X\alpha. \tag{21}$$

Note that here the inference can be conducted with matrix computation, which is different from that in conventional CRF. The reason is that in Continuous CRF the output variable is *continuous*, while in conventional CRF it is *discrete*.

If we ignore the relation between documents and set $\beta = 0$, then the ranking model degenerates to

$$\hat{y} = X\alpha,$$

which is equivalent to a linear model used in conventional local ranking.

For n documents, the time complexity of straightforwardly computing the ranking model (21) is of order $O(n^3)$ and thus it is expensive. The main cost of the computation comes from matrix inversion. In this paper we employ a fast computation technique to quickly perform the task. First, we make S a sparse matrix, which has at most K non-zero values in each row and each column. We can do so by only considering the similarity between each document and its $\frac{K}{2}$ nearest neighbors. Next, we use the Gibbs-Poole-Stockmeyer algorithm [17] to convert S to a banded matrix. Finally we solve the following system of linear equation and take the solution as ranking scores.

$$(\alpha^T eI + \beta D - \beta S)\hat{y} = X\alpha \tag{22}$$

[1]For example, if $X = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$, then $X := [1, 2, 3, 4]^T$.

[2]Stochastic Gradient means conducting gradient ascent from one instance to another. In our case, an instance corresponds to a query.

Let $A = \alpha^T eI + \beta(D - S)$. A is a banded matrix when S is a banded matrix. Then, the scores $\hat{y}$ in Eq.(22) can be computed with time complexity of $O(n)$ when $K \ll n$ [9]. That is to say, the time complexity of testing a new query is comparable with those of existing local ranking methods.

5.2 Topic Distillation

We can also specify Continuous CRF to make it suitable for Topic Distillation. Here we assume that the parent-child relation between two pages is given. The input vectors are also defined. What we need to do is to specify the feature functions.

We define the vertex feature function $f_k(y_i, x)$ in the same way as that in Eq.(8). Furthermore, we define the only edge feature function as

$$g(y_i, y_j, x) = R_{i,j}(y_i - y_j), \tag{23}$$

where $R_{i,j}$ denotes the parent-child relation: $R_{i,j} = 1$ if document i is the parent of j, and $R_{i,j} = 0$ for other cases.

The Continuous CRF for Topic Distillation then becomes

$$\Pr(y|x) = \frac{1}{Z(x)} \exp\left\{ \sum_i \sum_{k=1}^{K1} -\alpha_k (y_i - x_{i,k})^2 + \sum_{i,j} \beta R_{i,j}(y_i - y_j) \right\}, \tag{24}$$

where $Z(x)$ is defined as

$$Z(x) = \int_y \exp\left\{ \sum_i \sum_{k=1}^{K1} -\alpha_k (y_i - x_{i,k})^2 + \sum_{i,j} \beta R_{i,j}(y_i - y_j) \right\} dy. \tag{25}$$

To guarantee that $\exp\left\{ \sum_i \sum_{k=1}^{K1} -\alpha_k (y_i - x_{i,k})^2 + \sum_{i,j} \beta R_{i,j}(y_i - y_j) \right\}$ is integrable, we require that $\alpha_k > 0$.

The Continuous CRF can naturally model Topic Distillation. First, if the value of $x_{i,k}$ is high, then the value of y_i is high with high probability. Second, if the value of $R_{i,j}$ is one, then the value of y_i is large than that of y_j with high probability.

With some derivation, we have

$$Z(x) = (2a)^{-\frac{n}{2}} (2\pi)^{\frac{n}{2}} \exp(\frac{1}{4a} b^T b - c), \tag{26}$$

where n is the number of documents for this query, and $a = \alpha^T e$, $b = 2X\alpha + \beta(D_r - D_c)e$, D_r and D_c are two diagonal matrixes with $D_{ri,i} = \sum_j R_{i,j}$ and $D_{ci,i} = \sum_j R_{j,i}$, $c = \sum_i \sum_{k=1}^{K1} \alpha_k x_{i,k}^2$.

In learning, we use Gradient Ascent to maximize the log likelihood. Again, we use the technique for optimization to guarantee $\alpha_k > 0$. We compute the derivative of $L(\theta)$ with respect to β and the new optimization variable $\log \alpha_k$ as follows.

$$\nabla_{\log \alpha_k} = \frac{\partial L(\theta)}{\partial \log \alpha_k} = \alpha_k \sum_{q=1}^{N} \left\{ \sum_i -(y_i^{(q)} - x_{i,k}^{(q)})^2 - \frac{\partial \log Z(x^{(q)})}{\partial \alpha_k} \right\} \tag{27}$$

$$\nabla_\beta = \frac{\partial L(\theta)}{\partial \beta} = \sum_{q=1}^{N} \left\{ \sum_{i,j} R_{i,j}^{(q)} (y_i^{(q)} - y_j^{(q)}) - \frac{\partial \log Z(x^{(q)})}{\partial \beta} \right\} \tag{28}$$

Now we show how to get the partial derivative $\frac{\partial \log Z(x^{(q)})}{\partial \alpha_k}$ and $\frac{\partial \log Z(x^{(q)})}{\partial \beta}$. For simplicity, we omit the super script (q) hereafter.

$$\frac{\partial \log Z(x)}{\partial \alpha_k} = -\frac{n}{2a} - \frac{1}{4a^2} b^T b + \frac{1}{2a} b^T X_{,k} - \sum_i x_{i,k}^2 \tag{29}$$

$$\frac{\partial \log Z(x)}{\partial \beta} = \frac{1}{2a} b^T (D_r - D_c) e \tag{30}$$

where $X_{,k}$ denotes the k-th column of matrix X.

Substituting Eq. (29) and (30) into Eq. (27) and (28), we obtain the gradient of the log likelihood function. Here we omit the details of the learning algorithm, which is similar to Algorithm 1.

In inference, we calculate the ranking scores of documents with respect to a new query in the following way.

$$\hat{y} = \arg\max_y \Pr(y|x;\theta) = \frac{1}{\alpha^T e} (2X\alpha + \beta(D_r - D_c)e) \tag{31}$$

Similarly to Pseudo Relevance Feedback, if we ignore the relation between documents and set $\beta = 0$, the ranking model degenerates to a linear ranking model in conventional local ranking.

5.3 Combination

We can also conduct multiple global ranking tasks simultaneously. For example, we can combine Pseudo Relevance Feedback and Topic Distillation by using the following Continuous CRF model

$$\begin{aligned} \Pr(y|x) &= \frac{1}{Z(x)} \exp\left\{ \sum_i \sum_{k=1}^{K1} -\alpha_k (y_i - x_{i,k})^2 \right. \\ &+ \left. \sum_{i,j} \left(\beta_1 R_{i,j} (y_i - y_j) - \beta_2 \frac{S_{i,j}}{2} (y_i - y_j)^2 \right) \right\}. \end{aligned}$$

In this case, the ranking scores of documents for a new query is calculated as follows.

$$\hat{y} = \arg\max_y \Pr(y|x;\theta) = (\alpha^T e I + \beta_2 D - \beta_2 S)^{-1} \left(X\alpha + \frac{\beta_1}{2} (D_r - D_c) e \right)$$

Continuous CRF is a powerful model in the sense that various types of relations can be incorporated as edge feature functions.

6 Experiments

We applied Continuous CRF to Pseudo Relevance Feedback and Topic Distillation. We also compared the performances of Continuous CRF model against several baseline methods in the two tasks. As data, we used LETOR [18], which is a dataset for

learning to rank research[3]. We made use of OHSUMED in LETOR for Pseudo Relevance Feedback and TREC in LETOR for Topic Distillation. As evaluation measure, we utilized NDCG@n (Normalized Discounted Cumulative Gain) [12].

6.1 OHSUMED: Pseudo Relevance Feedback

6.1.1 Data Set

The OHSUMED dataset in LETOR is derived from the OHSUMED data for research on *relevance search* [11]. The document collection is a subset of MEDLINE, a database on medical publications.

There are 106 queries in OHSUMED data set, each associated with a number of documents. The relevance degrees of documents with respect to the queries are judged by humans, on three levels: *definitely relevant*, *partially relevant*, or *not relevant*. There are in total 16,140 query-document pairs with relevance judgments. Each query-document pair is represented by a 25 dimension feature vector, which means $K1 = 25$ in Eq. (10). Details of features can be found in [18].

Similarity between documents is provided as relation information in the data, which is defined in the following way. First stop words are removed from the documents. Each document is represented as a term vector in the vector space model [1]. Then the similarity $S_{i,j}$ between two documents i and j is calculated as cosine between the term vectors of the two documents. (One should not confuse the term vector with the input vector in learning.)

6.1.2 Baseline Methods

As baseline methods, we used RankSVM [10] and ListNet [3]. RankSVM is a state-of-the-art algorithm of the pairwise approach to learning to rank, and ListNet is a state-of-the-art algorithm of the listwise approach.

The two learning to rank methods only use content information, but not relation information. To make fair comparisons, we added post processing to the two methods in which we incorporated relation information into the final ranking model creation. We refer to the methods as 'RankSVM plus relation' (RankSVM+R) and 'ListNet plus relation' (ListNet+R). Following the work in [8], we first calculated the scores of documents based on content information using a learning to rank method (RankSVM or ListNet), then propagated the scores using similarities between documents, and finally ranked the documents based on the propagated scores. Specifically, the final score list is calculated as

$$y_{+r} = (I + \beta(D - S))^{-1} y,$$

where y is the score list output by a learning algorithm (e.g. RankSVM or ListNet), and D and S are the same as those in Eq.(12). Here β is a parameter balancing the influence from content information and relation information.

For reference purposes, we also tried BM25 and Pseudo Relevance Feedback based on BM25. For the two baselines, we used the tools provided in Lemur toolkit[4].

[3]The data is available at http://research.microsoft.com/users/LETOR/.

[4]http://www.lemurproject.org/

Table 1: Ranking Accuracy on OHSUMED

Algorithms	ndcg1	ndcg2	ndcg3	ndcg5	ndcg10
BM25	0.3994	0.3931	0.3939	0.3972	0.3967
PRF	0.3962	0.4277	0.4104	0.3981	0.3925
RankSVM	0.4952	0.4755	0.4649	0.4579	0.4411
ListNet	0.5231	0.497	0.4777	0.4662	0.4489
RankSVM+R	0.5143	0.4676	0.462	0.4593	0.4431
ListNet+R	0.5391	0.4946	0.4663	0.4555	0.4308
CRF	**0.5443**	**0.4986**	**0.4881**	**0.4808**	**0.4537**

6.1.3 Experimental Results

We conducted 5 fold cross validation for Continuous CRF and all the baseline methods, using the partition provided in LETOR.

Continuous CRF needs to use queries, their associated documents, and relevance scores as training data. Since LETOR only provides relevance labels, we mapped the labels to scores using heuristics. The rule is the ground truth score of a relevant document should be larger than that of an irrelevant document. We used validation set in LETOR to select the best mapping function.

For RankSVM+R and ListNet+R, we ran a number of experiments based on different values of parameter β. Here, we report the best performances of the methods. For RankSVM+R, $\beta = 0.2$; for ListNet+R, $\beta = 0.1$.

Table 1 shows the ranking accuracies of BM25, BM25 based Pseudo Relevance Feedback (PRF), RankSVM, ListNet, RankSVM+R, ListNet+R, and Continuous CRF (CRF), in terms of NDCG averaged over five trials.

CRF's performance is superior to the performances of RankSVM and ListNet. This is particularly true for NDCG@1; CRF achieves about 5 points higher accuracy than RankSVM and more than 2 points higher accuracy than ListNet. The results indicate that learning with similarity information can indeed improve search relevance.

We can see that CRF performs much better than RankSVM+R and ListNet+R at all NDCG positions. This indicates that with the same information the proposed CRF can indeed perform better than the heuristic baseline methods.

RankSVM+R beats RankSVM largely at NDCG@1, while obtains similar results at NDCG@3-10, but a worse result at NDCG@2. ListNet+R works better than ListNet at NDCG@1, but does not as well as ListNet at the other positions. This seems to indicate that heuristically using relation in post-processing does not work well.

Continuous CRF also outperforms PRF (Pseudo Relevance Feedback), the traditional method of using similarity information for ranking. The result suggests that it is better to leverage the machine learning techniques in Pseudo Relevance Feedback.

We made analysis on the results and found that CRF can indeed improve relevance. Table 2 and 3 show the top 10 results of RankSVM and CRF for query "back pain-mri sensitivity etc in comparison to ct of lumbar spine" respectively. The documents in red are 'definitely relevant', documents in blue are 'partially relevant', and documents in black are 'not relevant'.

Table 2: Top 10 Results of RankSVM

Doc ID	Title
184771	The pseudoradicular syndrome. Lower extremity peripheral nerve entrapment masquerading as lumbar radiculopathy.
188516	Infections caused by central venous catheters in patients with acquired immunodeficiency syndrome.
277424	A randomized double-blind prospective study of the efficacy of pulsed electromagnetic fields for interbody lumbar fusions.
169405	No clinical effect of back schools in an HMO. A randomized prospective trial.
93130	The relationship between leg length discrepancy and lumbar facet orientation.
217695	Lumbar intraspinal synovial cysts. Recognition and CT diagnosis [see comments]
171218	Pain provocation and disc deterioration by age. A CT/discography study in a low-back pain population.
189343	Metaplastic proliferative fibrocartilage as an alternative concept to herniated intervertebral disc.
255894	Thoracic and lumbar spine trauma.
202875	Macroamylasemia: a simple stepwise approach to diagnosis [see comments]

It is obvious that CRF works better than RankSVM for this query. Document 262357 is a 'definitely relevant' document, but is ranked out of top 10 by RankSVM. Since this document is similar to documents 277424 and 169405 which are ranked at position 2 and 5, it is boosted to position 6 by CRF using similarity information.

6.2 TREC: Topic Distillation

6.2.1 Data Set

In TREC 2004, there was a track for web search, called Topic Distillation, which is aimed at enhancing research on Topic Distillation, the task described in Section 2.

The TREC dataset in LETOR is derived from the TREC 2004 Topic Distillation data. There are 75 queries, and each query associated with about 1,000 documents. Each query document pair is associated with a label, representing whether the document is an entry page to the query (an answer) or not. There are 44 features defined over a query-document pair (refer to [18] for the details). It implies that $K1 = 44$ in Eq.(24). Furthermore, information on parent-child relation is also given. The element

Table 3: Top 10 Results of CRF

Doc ID	Title
184771	The pseudoradicular syndrome. Lower extremity peripheral nerve entrapment masquerading as lumbar radiculopathy.
277424	A randomized double-blind prospective study of the efficacy of pulsed electromagnetic fields for interbody lumbar fusions.
188516	Infections caused by central venous catheters in patients with acquired immunodeficiency syndrome.
189343	Metaplastic proliferative fibrocartilage as an alternative concept to herniated intervertebral disc.
169405	No clinical effect of back schools in an HMO. A randomized prospective trial.
262357	Intramuscular depot methylprednisolone induction of chrysotherapy in rheumatoid arthritis: a 24-week randomized controlled trial.
249197	A prospective study of nerve root infiltration in the diagnosis of sciatica. A comparison with radiculography, computed tomography, and operative findings.
254602	The neuroradiographic diagnosis of lumbar herniated nucleus pulposus: II. A comparison of computed tomography (CT), myelography, CT-myelography, and magnetic resonance imaging.
217695	Lumbar intraspinal synovial cysts. Recognition and CT diagnosis [see comments]
255894	Thoracic and lumbar spine trauma.

$R_{i,j}$ equals 1 if page i is parent of page j in a website, and equals 0 otherwise [5].

6.2.2 Baseline Methods

As baseline methods, we used RankSVM [10] and ListNet [3].

Since RankSVM and ListNet do not use relation information, we tried two modifications of them, in which we used relation information in post processing. We refer to them as 'RankSVM plus relation' (RankSVM+R) and 'ListNet plus relation' (ListNet+R). In RSVM+R (or ListNet+R), we use the ranking score of a child page output by RankSVM (or ListNet) to enhance the ranking score of its parent also output by

[5] This can be mined from the URL hierarchy [25].

RankSVM (or ListNet). The idea is similar to that of sitemap based score propagation [24].

We also tested non-learning methods of BM25 and sitemap based relevance propagation [24]. The basic idea of sitemap based relevance propagation is to use the relevance of a child page to enhance the relevance of its parent page. There are two variants of the method: sitemap based term propagation ('ST' for short) and sitemap based score propagation ('SS' for short).

6.2.3 Experimental Results

We conduced 5-fold cross validation on our method and the baseline methods, using the partitions in LETOR. For RankSVM and ListNet, we refer to the results in LETOR.

Continuous CRF needs to use queries, their associated documents, and relevance scores as training data, while LETOR only provides ranking labels. Again, we used heuristics to map ranking labels to ranking score. The rule is that the score of an answer document should be larger than that of a non-answer document. We used the validation set in LETOR to select the best mapping function.

Table 4 shows the performances of BM25, SS, ST, RankSVM, ListNet, and CRF model in terms of NDCG averaged over 5 trials.

CRF outperforms RankSVM and ListNet at all NDCG positions. This is particularly true for NDCG@1. CRF achieves 8 points higher accuracy than RankSVM and ListNet, which is a more than 15% relative improvement. Overall, learning using relation information can achieve better results than learning without using relation information. The result indicates that our method can effectively use the information in training of a Topic Distillation model.

CRF performs much better than RankSVM+R and ListNet+R at all NDCG positions. This indicates that with the same information the proposed CRF can indeed perform better than the heuristic methods.

RankSVM+R beats RankSVM largely at NDCG@1, while obtains slightly better results at NDCG@3-10 but a slightly worse result at NDCG@2. ListNet+R works better than ListNet at NDCG@2-10, but does not at NDCG@1. The results seem to indicate that simply using relation information as post-processing does not work very well.

Continuous CRF also outperforms SS and ST, the traditional method of using parent-child information for Topic Distillation. The result suggests that it is better to leverage the machine learning techniques using both content information and relation information in ranking.

We investigated the reason that CRF can achieve better results than other algorithms and concluded that it is because CRF can successfully leverage the relation information in ranking. Without loss of generality, we make a comparison between CRF and RankSVM.

Table 5 show top 10 results of RankSVM and CRF for query "HIV/AID". The answer pages for this query are in red color. The answer page 643908 is not ranked in top 10 by RankSVM, because its content feature is not strong. Since the content features of its child pages (such as 220602, 887722, and other pages) are very strong, CRF can effectively use the parent-child relation information and boost it to position 4.

Table 4: Ranking Accuracy on TREC2004

Algorithms	ndcg1	ndcg2	ndcg3	ndcg5	ndcg10
BM25	0.3067	0.2933	0.2578	0.2293	0.1747
ST	0.3200	0.3133	0.3111	0.3232	0.3452
SS	0.3200	0.3200	0.3130	0.3227	0.3440
RankSVM	0.4400	0.4333	0.4092	0.3935	0.4201
ListNet	0.4400	0.4267	0.4371	0.4209	0.4579
RankSVM+R	0.4933	0.4200	0.4118	0.4027	0.4197
ListNet+R	0.4400	0.4467	0.4481	0.4327	0.4591
CRF	**0.5200**	**0.4733**	**0.4552**	**0.4428**	**0.4604**

7 Conclusions

In search, not only content information of documents, but also relation information between documents are needed for ranking of documents. Not only relevance, but also importance and diversity need to be considered in ranking. Previously, the problems were addressed as separated issues. In this paper, we call the general problem as global ranking and have given a definition of it.

We have further proposed a learning to rank method for global ranking, using a Continuous CRF model. We have devised a learning method for creating Continuous CRF using training data. Taking Pseudo Relevance Feedback and Topic Distillation as examples, we have showed how to use Continuous CRF in global ranking. Experimental results on benchmark data show that our method using Continuous CRF improves upon the baseline methods for the two tasks.

There are still issues which we need to investigate at the next step. (1) We have assumed that in training each training document is assigned a ranking score. In practice, such training data would be difficult to obtain. One possibility is to use click through data. How to generate ranking scores from click through data is still an open problem. (2) We have studied the method of learning Continuous CRF with Maximum Likelihood Estimation. It is interesting to see how to apply Maximum A Posteriori Estimation to the problem as well. (3) We have studied two retrieval tasks: Pseudo Relevance Feedback and Topic Distillation. We plan to look at more tasks in the future. (4) We have applied CRF model to two global ranking tasks separately. We will investigate the combination of multiple global ranking tasks.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999.

[2] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML '05*, pages 89–96, 2005.

[3] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML '07*, pages 129–136, 2007.

Table 5: Top 10 Results of RankSVM and CRF

Doc ID	URL
	Results of RankSVM
54703	http://hab.hrsa.gov/
220602	http://www.cdc.gov/hiv/dhap.htm
887722	http://www.cdc.gov/hiv/hivinfo.htm
220487	http://www.fda.gov/oashi/aids/hiv.html
28888	http://www.usinfo.state.gov/topical/global/hiv/
521604	http://www.cdc.gov/hiv/pubs/brochure/atrisk.htm
862409	http://www.cdc.gov/hiv/hivaids.htm
390722	http://www.hud.gov/offices/cpd/aidshousing/index.cfm
219192	http://www.niaid.nih.gov/newsroom/.../default.htm
454764	http://www.cdc.gov/hiv/graphics/adolesnt.htm
	Results of CRF
54703	http://hab.hrsa.gov/
3477	http://www.hiv.omhrc.gov/
220602	http://www.cdc.gov/hiv/dhap.htm
643908	http://www.cdc.gov/hiv/index.htm
28888	http://www.usinfo.state.gov/topical/global/hiv/
334549	http://www.surgeongeneral.gov/aids/default.htm
45756	http://www.metrokc.gov/health/apu/
219192	http://www.niaid.nih.gov/newsroom/.../default.htm
321364	http://www.metrokc.gov/health/apu/epi/index.htm
547722	http://www.usaid.gov/pop_health/.../index.html

[4] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.

[5] W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35(4):285–295, 1979.

[6] F. Diaz. Regularizing ad hoc retrieval scores. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 672–679, New York, NY, USA, 2005. ACM Press.

[7] F. Diaz. Regularizing query-based retrieval scores. *Information Retrieval*, 10(6):531–562, December 2007.

[8] F. Diaz. Regularizing query-based retrieval scores. *Information Retrieval*, 2007.

[9] G. H. Golub and C. F. V. Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[10] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *ICANN1999*, pages 97–102, 1999.

[11] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *SIGIR '94*, pages 192–201, 1994.

[12] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[13] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, pages 133–142, 2002.

[14] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[15] K. L. Kwok. A document-document similarity measure based on cited titles and probability theory, and its application to relevance feedback retrieval. In *SIGIR '84*, pages 221–231, 1984.

[16] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01*, pages 282–289, 2001.

[17] J. G. Lewis. Algorithm 582: The gibbs-poole-stockmeyer and gibbs-king algorithms for reordering sparse matrices. *ACM Trans. Math. Softw.*, 8(2):190–194, 1982.

[18] T.-Y. Liu, J. Xu, T. Qin, W.-Y. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR '07 Workshop*, 2007.

[19] A. McCallum. Efficiently inducing features of conditional random fields. In *UAI03*, 2003.

[20] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *HLT-NAACL '03*, pages 188–191, 2003.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[22] D. Pinto, A. McCallum, X. Wei, and W. B. Croft. Table extraction using conditional random fields. In *SIGIR '03*, pages 235–242, 2003.

[23] T. Qin, T.-Y. Liu, W. Lai, X.-D. Zhang, D.-S. Wang, and H. Li. Ranking with multiple hyperplanes. In *SIGIR '07*, pages 279–286, 2007.

[24] T. Qin, T.-Y. Liu, X.-D. Zhang, Z. Chen, and W.-Y. Ma. A study of relevance propagation for web search. In *SIGIR '05*, pages 408–415, 2005.

[25] T. Qin, T.-Y. Liu, X.-D. Zhang, G. Feng, D.-S. Wang, and W.-Y. Ma. Topic distillation via sub-site retrieval. *Information Processing & Management*, 43(2):445–460, 2007.

[26] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *NAACL '03*, pages 134–141, 2003.

[27] A. Shakery and C. Zhai. A probabilistic relevance propagation model for hypertext retrieval. In *CIKM2006*, pages 550–558, 2006.

[28] M. D. Smucker and J. Allan. Find-similar: similarity browsing as a search tool. In *SIGIR '06*, pages 461–468, 2006.

[29] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006. To appear.

[30] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *SIGIR '06*, pages 162–169, 2006.

[31] M.-F. Tsai, T.-Y. Liu, T. Qin, H.-H. Chen, and W.-Y. Ma. Frank: a ranking method with fidelity loss. In *SIGIR '07*, pages 383–390, 2007.

[32] E. Voorhees and D. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.

[33] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR '07*, pages 271–278, 2007.

[34] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR '03*, pages 10–17, 2003.

[35] Z. Zheng, K. Chen, G. Sun, and H. Zha. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR '07*, pages 287–294, 2007.

[36] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. Simultaneous record detection and attribute labeling in web data extraction. In *KDD '06*, pages 494–503, 2006.