Top-k Learning to Rank: Labeling, Ranking and Evaluation

Shuzi Niu, Jiafeng Guo, Yanyan Lan, Xueqi Cheng
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P.R. China
niushuzi@software.ict.ac.cn, {guojiafeng, lanyanyan, cxq}@ict.ac.cn

ABSTRACT

In this paper, we propose a novel top-k learning to rank framework, which involves labeling strategy, ranking model and evaluation measure. The motivation comes from the difficulty in obtaining reliable relevance judgments from human assessors when applying learning to rank in real search systems. The traditional absolute relevance judgment method is difficult in both gradation specification and human assessing, resulting in high level of disagreement on judgments. While the pairwise preference judgment, as a good alternative, is often criticized for increasing the complexity of judgment from $\mathcal{O}(n)$ to $\mathcal{O}(n \log n)$. Considering the fact that users mainly care about top ranked search results, we propose a novel top-k labeling strategy which adopts the pairwise preference judgment to generate the top k ordering items from n documents (i.e. top-k ground-truth) in a manner similar to that of HeapSort. As a result, the complexity of judgment is reduced to $\mathcal{O}(n \log k)$. With the top-k ground-truth, traditional ranking models (e.g. pairwise or listwise models) and evaluation measures (e.g. NDCG) no longer fit the data set. Therefore, we introduce a new ranking model, namely FocusedRank, which fully captures the characteristics of the top-k ground-truth. We also extend the widely used evaluation measures NDCG and ERR to be applicable to the top-k ground-truth, referred as κ-NDCG and κ-ERR, respectively. Finally, we conduct extensive experiments on benchmark data collections to demonstrate the efficiency and effectiveness of our top-k labeling strategy and ranking models.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms, Performance, Experimentation, Theory

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

Keywords

Learning to Rank, Top-k, Preference Judgment, Evaluation

1. INTRODUCTION

In the past few years, learning to rank has been widely recognized as an important technique for information retrieval (IR). A vital part to employ learning to rank in real search systems is the acquisition of reliable and high quality labeled datasets, both for training and evaluation. In traditional IR literature, assessors are requested to determine the relevance of a document under some pre-defined gradations, which is called *absolute relevance judgment* method. However, there are some significant drawbacks for this evaluation process. Firstly, the specifics of the gradations (i.e. how many grades to use and what those grades mean) must be defined, and it is not clear how these choices will affect relative performance measurements [26]. Secondly, the assessing burden increases with the complexity of the relevance gradations; the choice of label is not clear when there are more factors to consider, leading to high level of disagreement on judgments [4].

Recently *pairwise preference judgment* has been investigated as a good alternative [20, 26]. Instead of assigning a relevance grade to a document, an assessor looks at two pages and judges which one is better. Compared with absolute relevance judgment, the advantages lie in that: (1) There is no need to determine the gradation specifications as it is a binary decision. (2) It is easier for an assessor to express a preference for one document over the other than to assign a pre-defined grade to each of them [7]. (3) Most state-of-the-art learning to rank models, pairwise or listwise, are trained over preferences. As noted by Carterette et al. [7], "by collecting preferences directly, some of the noise associated with difficulty in distinguishing between different levels of relevance may be reduced." Although preference judgment likely produce more reliable labeled data, it is often criticized for increasing the complexity of judgment (e.g. from $\mathcal{O}(n)$ to $\mathcal{O}(n \log n)$ [20]), which poses a big challenge in wide use. Do we actually need to judge so many pairs for real search systems? If not, which pairs do we choose? How to choose? These questions become the original motivation of this paper.

As we know, in real Web search scenario, it is well accepted that users mainly care about the top results [30]. In other words, the ordering of the top results (typically the results on the first one or two pages) is critical for users' search experience. It indicates that a labeling strategy shall take effort to figure out the top results and judge the preference orders among them, but pay less attention to the exact

preference orders among the rest results. Based on this observation, we propose a novel top-k labeling strategy which adopts the pairwise preference judgment to generate the top k ordering items from a set of n items in a manner similar to that of HeapSort. The obtained ground-truth from this top-k labeling strategy is a mixture of the total order of the top k items, and the relative preferences between the set of top k items and the set of the rest $n-k$ items, referred as top-k ground-truth. With this top-k labeling strategy, we can not only capture enough information for learning to rank [30], but also largely reduce the complexity of judgment to $\mathcal{O}(n \log k)$.

With top-k ground-truth, we find that traditional ranking models, either pairwise or listwise, are no longer suitable for the labeled data set. It is natural to introduce a mixed ranking model, with the listwise model capturing the total order of the top k items and the pairwise model capturing the relative preference between the set of top k items and the set of the rest $n-k$ items. Such a mixed model can thus combine the advantages of both pairwise and listwise approaches to fully exploit the information in the top-k ground-truth. We refer such a mixed ranking model as FocusedRank, since it emphasizes more on the ordering of the top items.

For evaluation, traditional IR evaluation measures (e.g., MAP, NDCG and ERR), which are mainly defined on the absolute judgment, cannot be directly applied to the top-k ground-truth. To address this problem, we extend NDCG and ERR to κ-NDCG and κ-ERR by taking a function of the position of items as the absolute relevance label. The proposed evaluation measures thus emphasize the importance of the ordering of the top k items. Unlike the evaluation measures based on preference judgments [6], κ-NDCG and κ-ERR keep the same form as NDCG and ERR thus enjoy all the merits of traditional IR evaluation measures.

Finally, we conduct extensive experiments on benchmark data collections. Major experimental findings include: (1) With top-k labeling strategy, the time cost on labeling one pair is much less than that on one item in absolute relevance judgment, and the overall time cost is comparable with that in absolute relevance judgment. (2) With top-k labeling strategy, the level of agreement on judgments is higher than that on absolute relevance judgment. (3) With FocusedRank, the ranking performance is significantly better than the state-of-the-art pairwise and listwise ranking models.

To sum up, we propose a top-k learning to rank framework[1], a novel and complete framework including labeling strategy, ranking model and evaluation measures. Our main contributions are as follows:

1. We propose a novel top-k labeling strategy which adopts pairwise preference judgment to obtain reliable ground-truth for learning to rank. As a result, the complexity of judgment is reduced to $\mathcal{O}(n \log k)$.

2. We introduce a new ranking model named FocusedRank to capture the characteristics of the top-k ground-truth, and it outperforms the state-of-the-art pairwise and listwise ranking models on benchmark datasets.

3. We derive two new evaluation measures named κ-NDCG and κ-ERR applicable to the top-k ground-truth. They both emphasize the importance of top-k ordering and enjoy the merits of traditional IR measures with the similar formulation.

[1]Note that Xia et al. also mentioned the top k ranking problem in [30]. The difference is that they focus on the ranking models under the circumstances of traditional labeling strategy and evaluation measure.

2. RELATED WORK

In this section, we briefly review some related work on labeling strategy, ranking model and evaluation measure in learning to rank literature.

2.1 Labeling Strategy

In learning to rank, labeling strategies can be divided into two categories: absolute judgment and relative judgment [21, 26, 32, 7].

In absolute judgment, assessors are usually requested to assign a graded score to an item independent of the other items [15, 4, 27, 28, 29] under some pre-defined gradations. Such a labeling strategy has been widely adopted in both industry and academia to construct benchmark datasets in IR, e.g. TREC data sets (since 2000), Microsoft learning to rank datasets [18] and Yahoo! Learning to Rank challenge 2010 data set. One difficult problem in using absolute judgment is to clearly define the specifics of the gradations, i.e. how many grades to use and what those grades mean. Some previous studies tried to figure out the proper number of relevance gradations [22, 19]. However, as noted by Cox [12], "there is no single number of response alternatives for a scale which is appropriate under all circumstances". If the descriptions of each degree are not clearly defined, a multi-grade judgment method can be easily misused in the evaluation process [34]. Moreover, the assessing burden in absolute judgment increases with the complexity of the relevance gradations. When there are more factors to consider, the choice of label is not clear, resulting in high level of disagreement on judgments [4].

In contrast, relative judgement aims to directly judge the relative order of a set of items [26]. As a typical form of relative judgment, pairwise preference judgment asks assessors to express a preference for one item over the other [6]. One concern of using this strategy is the complexity of judgment since the number of item pairs is polynomial in the number of items. Carterette et al. [7] attempted to reduce the number of pairs for judging by using transitivity of relevance among documents. Nir Ailon [1] proposed a formal pairwise method based on QuickSort which can reduce the number of preference judgments from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$. Compared with $\mathcal{O}(n)$ in absolute judgment, this is still not affordable for assessors. To increase the efficiency of relative judgment, R. Song et al. [26] further proposed to select the best one each time from the remaining items, which is only applicable to small datasets. Different from the above related work, we propose a novel top-k labeling strategy to largely save the effort of preference judgment, by exploiting what Web search users actually care about on ranking.

2.2 Ranking Model

So far learning to rank has been mainly addressed by pointwise, pairwise, and listwise ranking models. In pointwise models [17], ranking is transformed to regression or classification on individual items to represent the absolute label on each item. In pairwise models [14, 11, 3], ranking is transformed to classification on item pairs to represent the

preference between two items. In listwise models [33, 31, 5], instances as document lists are generated through the comparison over item pairs, and it is superior in modeling more discriminative judgments. Therefore, we can conclude that, pointwise models is well suited for absolute relevance judgment; while both pairwise and listwise models are applicable in either absolute or relative judgment scenario.

In previous work [30], Xia et al. extended three listwise ranking models, namely top-k ListMLE, top-k ListNet and top-k RankCosine, to fit the top-k scenario. Note that they addressed this scenario under the circumstances of traditional labeling strategy and evaluation measure. They conducted experiments on the top-k ListMLE, and claimed that the top-k ListMLE can outperform traditional pairwise and listwise ranking models. However, it cannot avoid the computational complexity on the entire permutation such as that in top-k ListNet.

In our paper, we take the above ranking models as the baselines to show the superiority of FocusedRank.

2.3 Evaluation Measure

To evaluate the effectiveness of a ranking model, many IR measures have been proposed. Here we give a brief introduction to several popular ones which are widely used in learning to rank. See also [16] for other measures.

Precision@k [2] is a measure for evaluating top k positions of a ranked list using two grades (relevant and irrelevant) of relevance judgment. With Precision as the basis, Average Precision (AP) and Mean Average Precision (MAP) [2] are derived to evaluate the average performance of a ranking model.

While Precision considers only two graded relevance judgments, Discounted Cumulated Gain (DCG) [13] is an evaluation measure that can leverage the relevance judgment in terms of multiple ordered categories, and has an explicit position discount factor in its definition. By normalizing DCG@k with its maximum possible value, we will get another popular measure named Normalized Discounted Cumulated Gain (NDCG).

To relax the additive nature and the underlying independence assumption in NDCG, another evaluation measure, Expected Reciprocal Rank (ERR), is proposed in [8]. It implicitly discounts documents which are shown below very relevant documents, and is defined as the expected reciprocal length of time that the user will take to find a relevant document.

Although MAP, NDCG and ERR are widely used in IR, they all adopt absolute relevance labels in their formulation, which imposes restrictions on direct application in the scenario of relative judgment. Therefore, new measures such as $bpref$, $ppref$, and $nwppref$ [6] have been proposed. However, these measures have not been widely accepted by IR community. In this paper, we extend traditional IR evaluation measures to our relative judgment scenario with similar formulation.

3. TOP-K LEARNING TO RANK

In this section, we will introduce our top-k learning to rank framework in detail, which involves the labeling strategy, the ranking model and the evaluation measure.

3.1 Top-k Labeling Strategy

According to previous work and the above discussions, pairwise preference judgment is superior to traditional absolute relevance judgment in the acquisition of reliable judgments from human assessors. However, it is often criticized for increasing the complexity of judgment. In this section, we propose a novel top-k labeling strategy by exploiting what Web search users actually care about on ranking. Our labeling strategy can not only capture enough information for learning to rank, but also largely save the effort of preference judgment.

3.1.1 Motivation

In real Web search applications, users usually pay more attention to the top-k items [9, 25, 10]. For example, according to a user study [30], in modern search engines, about 62% of search users only click on the results within the first pages, and 90% of search users click on the results within the first three pages. It shows that the ordering of the top k results is critical for users' search experience. Two ranked lists of results will likely provide the same value to users (and thus suffer the same loss), if they have the same ranking results for the top positions [30]. Moreover, a good ranking on the top results is much more important than a good ranking on the others. Therefore, a labeling strategy shall take effort to figure out the top k results, judge the preference orders among them carefully, but pay less attention to the exact preference orders among the rest results.

3.1.2 Labeling Strategy

Based on the above analysis, we propose a novel top-k labeling strategy using the pairwise preference judgment as the basis. The basic assumption for our labeling strategy is the transitivity of preference judgments of relevance [24, 7]. That is, if i is preferred to j and j is preferred to k, the assessor will also prefers i to k. With this assumption, our labeling strategy generate the top k ordering items from a set of n items in a manner similar to that of HeapSort. It mainly takes the following three steps:

Step1 Randomly select k items from the set of n items and build a min-heap with t as the root based on the pairwise preference judgments by assessors. Here a min-heap is a complete binary tree with the property: if B is a child node of A, A is less relevant than B.

Step2 Randomly select an item r from the rest $n-k$ items and the preference judgement is conducted between t and r by assessors. We then update the heap if necessary and obtain a new min-heap with the k most relevant items up till now. It is repeated until all the items have been selected.

Step3 Sort the final k items in the min-heap in a descending order, and append the rest items after the k items.

The detailed labeling algorithm is shown in Algorithm1. With the above top-k labeling strategy, the obtained ground-truth is a mixture of the total order of the top k items and the relative preference between the set of top k items and the set of the rest $n-k$ items, referred to as top-k ground-truth.

3.1.3 Complexity Analysis

Here we consider the judgment complexity of each step in top-k labeling strategy:

(1)The judgment complexity of building a min-heap with k items in **Step 1** is $O(k)$;

Algorithm1: Top-k Labeling based on HeapSort

Input: (1) D, an item set; (2) k, top item number.
begin
  randomly select k items from D denoted as D_k.
  construct a min-heap H_k over D_k with preference judgment.
  for each item d in $D - D_k$ do
    obtain preference judgment over pair $(d, D_k[1])$.
    if the judgment is d more relevant than $D_k[1]$
      $D_k[1] = d$,
      update H_k over D_k with preference judgment.
    end if
  end for
  sort H_k to obtain top k items in descending order denoted as L_D.
  append $D - D_k$ to L_D.
end
Output: L_D.

(2)The judgment complexity in **Step 2** is $O((n-k)\log k)$ according to the complexity analysis for HeapSort;

(3)The judgment complexity in **Step 3** is $O(klogk)$.

Therefore, the total judgment complexity of top-k labeling strategy is about $O(n \log k)$. Compared with QuickSort strategy adopted by Nir Ailon [1] for preference judgment, our top-k labeling strategy significantly reduces the complexity from $O(n \log n)$ to $O(n \log k)$, where usually $k \ll n$. The judgment complexity of our strategy is nearly comparable with that of the absolute judgment (i.e. $O(n)$). Experimental results in the following section also verify the efficiency of our labeling strategy which is consistent with the theoretical analysis.

3.2 Top-k Ranking Model

With the top-k ground-truth obtained from our labeling method, traditional ranking models no longer fit the labeled dataset. On one hand, pairwise ranking models can capture the information of the relative preference, but fail to model the total order of the top k items since they ignore the position information. On the other hand, listwise ranking models can capture the information of the total order, but suffer from the great computational complexity due to a large undifferentiated item set in top-k ground-truth. To address this problem, we propose FocusedRank, a mixed ranking model with listwise ranking model capturing the total order of the top k items and pairwise ranking model capturing the relative preference between the set of top k items and the set of the rest $n-k$ items. Such a mixed model can thus combine the advantages of both pairwise and listwise approaches to fully exploit the information in the top-k ground-truth.

3.2.1 Notations

Given m training queries $\{q_i\}_{i=1}^m$, let $\mathbf{x}_i = \{x_1^{(i)}, \cdots, x_{n_i}^{(i)}\}$ be the items associated with q_i, where n_i is the number of documents of this query, T_i be the set of top k items and F_i be the set of other $n_i - k$ items. Denote the total order of T_i as a permutation π_i, where $\pi_i(x_j^{(i)})$ stands for the position of item $x_j^{(i)} \in T_i$. Denote $P_i = \{(x_u^{(i)}, x_v^{(i)}) : x_u^{(i)} \in T_i, x_v^{(i)} \in F_i\}$ as the set of pairs constructed between the set of top k items and the set of the rest $n-k$ items. We relate the top-k ground-truth to relevance labels by defining $\mathbf{y}_i = \{y_1^{(i)}, \cdots, y_{n_i}^{(i)}\}$ as position-aware relevance labels of the corresponding items, and $\mathbf{y}_i^{(T)} = \{y_j^{(i)} : x_j^{(i)} \in T_i.\}$. In our paper, we use $y_j^{(i)} = k + 1 - \pi_i(x_j^{(i)})$, if $x_j^{(i)} \in T_i$, and $y_j^{(i)} = 0$, otherwise. That is, suppose $k = 10$, the relevance labels for the top 10 items are defined in a descending order from 10 to 1, while the relevance labels for the rest items are defined as 0.

3.2.2 FocusedRank

In FocusedRank, we adopt a listwise loss to model the total order of top k items, and a pairwise loss to model the preference of top k items to the other items. The general loss function of FocusedRank on a query q_i is presented as follows[2].

$$L(f; q_i) = \beta \times L_{list}(f; T_i, \mathbf{y}_i) + (1-\beta) \times L_{pair}(f; P_i, \mathbf{y}_i), \quad (1)$$

where L_{list} stands for a listwise ranking model and L_{pair} stands for a pairwise ranking model, β is a trade-off coefficient to balance the two terms. As examples, we combine three popular listwise ranking models (i.e. SVM^{MAP}, AdaRank and ListNet) with three popular pairwise ranking models (i.e. RankSVM, RankBoost and RankNet) respectively to get three specific forms of FocusedRank, namely FocusedSVM, FocusedBoost and FocusedNet accordingly.

(1) FocusedSVM: RankSVM plus SVM^{MAP}

Both RankSVM [14] and SVM^{MAP} [33] apply the SVM technology to optimize the number of misclassified pairs and the average precision, respectively. Therefore, we combine these two ranking models together to get a new FocusedRank method, named FocusedSVM. Specifically, RankSVM is adopted to model the pairwise preference of P_i, and SVM^{MAP} is employed to model the total order of T_i. Therefore, L_{list} and L_{pair} in Eq. (1) has the following specific form, respectively.

$$L_{list} = \max_{\mathbf{z}_i^{(T)}} (1 - AP(\mathbf{z}_i^{(T)}, \mathbf{y}_i^{(T)}) + w^T \Psi(\mathbf{z}_i, T_i) - w^T \Psi(\mathbf{y}_i^{(T)}, T_i))$$

$$L_{pair} = \sum_{(x_u^{(i)}, x_v^{(i)}) \in P_i} \max\{0, 1 - (y_u^{(i)} - y_v^{(i)})(w^T x_u^{(i)} - w^T x_v^{(i)})\}.$$

Like RankSVM and SVM^{MAP}, FocusedSVM can then be formulated as an optimization problem as follows.

$$\min \frac{1}{2}\|w\|^2 + C \sum_{i=1}^m [\beta \zeta^{(i)} + (1-\beta) \sum_{(x_u^{(i)}, x_v^{(i)}) \in F_i} \xi_{u,v}^{(i)}]$$

$$s.t.: (y_u^{(i)} - y_v^{(i)})(w^T x_u^{(i)} - w^T x_v^{(i)}) \geq 1 - \xi_{u,v}^{(i)}, \forall (x_u^{(i)}, x_v^{(i)}) \in P_i,$$

$$w^T \Psi(\mathbf{y}_i^{(T)}, T_i) - w^T \Psi(\mathbf{z}_i, T_i) \geq 1 - AP(\mathbf{y}_i^{(T)}, \mathbf{z}_i^{(T)}) - \zeta^{(i)}, \forall \mathbf{z}_i^{(T)},$$

$$\xi_{u,v}^{(i)} \geq 0, \quad \zeta^{(i)} \geq 0, i = 1, ..., m,$$

where $\mathbf{z}_i^T$ stands for any incorrect label and Ψ is the same as that in SVM^{MAP}. Similar to SVM, $\frac{1}{2}\|w\|^2$ controls the complexity of the model w, and C is a trade-off parameter between the model complexity and hinge loss relaxations.

(2) FocusedBoost: RankBoost plus AdaRank

Both RankBoost [11] and AdaRank [31] adopt the boosting technology to output a ranking model by combining the week rankers, where the combination coefficients are determined by the probability distribution on document pairs and ranked lists respectively. Hence we combine these two

[2]In application, L_{list} and L_{pair} should be normalized to a comparable range, and we adopt this trick in our experiments.

Algorithm2: Learning Algorithm for FocusedBoost

Input: training data in terms of top-k ground-truth.
Given: initial distribution D_1 on T_i and D'_1 on P_i, i=1,$\cdots$,m.
For $t = 1, \cdots, T$
  train weak ranker f_t to minimize: $r_t = \beta \sum_{i=1}^{m} D_t(T_i) L_{list} + (1-\beta) \sum_{i=1}^{m} \sum_{(x_u^{(i)}, x_v^{(i)}) \in P_i} D'_t(x_u^{(i)}, x_v^{(i)}) L_{pair}$.
  choose $\alpha_t = \frac{1}{2} \log(\frac{1+r_t}{1-r_t})$.
  update
    $D_{t+1} = \frac{1}{Z_{t+1}} D_t(T_i) \exp(-E(\sum_{s=1}^{t} \alpha_s f_s, T_i, \mathbf{y}_i^T))$,
    $D'_{t+1} = \frac{1}{Z'_{t+1}} D'_t(x_u^{(i)}, x_v^{(i)}) \exp(\alpha_t (w^T x_u^{(i)} - w^T x_v^{(i)}))$,
    where,
    $Z_{t+1} = \sum_{i=1}^{m} D_t(T_i) \exp(-E(\sum_{s=1}^{t} \alpha_s f_s, T_i, \mathbf{y}_i^T))$.
    $Z'_{t+1} = \sum_{i=1}^{m} \sum_{(x_u^{(i)}, x_v^{(i)}) \in P_i} D'_t(x_u^{(i)}, x_v^{(i)}) \exp(\alpha_t (w^T x_u^{(i)} - w^T x_v^{(i)}))$.
Output: $f(x) = \sum_t \alpha_t f_t(x)$.

ranking models together to get a new FocusedRank method, named FocusedBoost. Specifically, RankBoost is adopted to model the preference of P_i, and AdaBoost is to model the total order of T_i. Therefore, L_{list} and L_{pair} in Eq. (1) has the following specific form, respectively.

$$L_{list} = \exp\left(-E(f, T_i, \mathbf{y}_i^T)\right),$$
$$L_{pair} = \exp\left(-(y_u^{(i)} - y_v^{(i)})(w^T(x_u^{(i)} - x_v^{(i)}))\right),$$

where $E(f, T_i, \mathbf{y}_i^{(i)})$ stands for the IR evaluation to optimize.

As in RankBoost and AdaRank, the detailed algorithm is shown in Algorithm2.

(3) FocusedNet: RankNet plus ListNet

Both RankNet and ListNet aim to optimize a cross entropy between the target probability and the modeled probability. The probability of the former is defined based on the exponential function of difference between the scores of any two documents in all document pairs given by the scoring function f. The probability of the latter is the permutation probability of a ranking list using Plackett-Luce model [23], which is also based on the exponential function. Hence we combine these two ranking models together to get a new FocusedRank method, named FocusedNet. Specially, RankNet is adopted to model the pairwise preference of P_i, and ListNet is to model the total order of T_i. Therefore, L_{list} and L_{pair} in Eq. (1) has the following specific form, specifically.

$$L_{list} = -\sum_{\sigma \in \Sigma_i} P_{\mathbf{y}_i^{(T)}}(\sigma) \log P_{\mathbf{z}_i^{(T)}}(\sigma),$$
$$L_{pair} = -\bar{P}_{u,v} \log P_{u,v}(f) - (1 - \bar{P}_{u,v}) \log(1 - P_{u,v}(f)),$$

where, $\bar{P}_{u,v} = 1$, if $y_u^{(i)} \geq y_v^{(i)}$, and $\bar{P}_{u,v} = 0$, otherwise. In addition, we have that

$$\mathbf{z}_i^{(T)} = \{z_j^{(i)} = w^T x_j^{(i)}, x_j^{(i)} \in T_i\},$$
$$P_{u,v}(f) = \frac{\exp(w^T x_u^{(i)} - w^T x_v^{(i)})}{1 + \exp(w^T x_u^{(i)} - w^T x_v^{(i)})},$$
$$P_s(\sigma) = \prod_{j=1}^{|T_i|} \frac{\phi(s_{\sigma(j)})}{\sum_{l=j}^{|T_i|} \phi(s_{\sigma(l)})}, s = \mathbf{y}_i^{(T)}, \mathbf{z}_i^{(T)},$$

where $s_{\sigma(j)}$ denotes the score of the object at position j of permutation σ.

3.3 Top-k Evaluation Measure

As aforementioned, traditional IR evaluation measures (e.g., MAP[3], NDCG and ERR) are mainly defined on the absolute judgment, thus cannot be directly applied to the top-k ground-truth. To address this problem, we extend NDCG and ERR to κ-NDCG and κ-ERR by taking the position of items as the absolute label using the way defined in Section 3.2.1. As a result, the derived evaluation measures actually emphasize the importance of the ordering of the top k items.

3.3.1 κ-NDCG

We first give the precise definition of NDCG as follows.

$$NDCG@l = \frac{1}{N_l} \sum_{j=1}^{l} \frac{2^{r_j} - 1}{\log_2(1+j)},$$

where r_j is the relevance label of the item with position j in the output ranked list, and N_l is a constant which denotes the maximum value of NDCG@l given the query.

Using the notations in Section 3.2.1, we can extend NDCG to κ-NDCG with the following definition.

$$\kappa - NDCG@l = \frac{1}{N'_l} \sum_{j=1}^{l} \frac{2^{y_j^{(i)}} - 1}{\log_2(1+j)}, \quad (2)$$

where N'_l is a constant which denotes the maximum value of κ-NDCG@l given the query.

3.3.2 κ-ERR

We first give the precise definition of ERR as follows.

$$ERR = \sum_{i=1}^{n} \frac{1}{n} R(r_i) \prod_{j=1}^{i-1} (1 - R(r_j)),\ R(r) = \frac{2^r - 1}{2^{r_{max}}},$$

where n is the document number of a query, and r_{max} is the highest relevance label in this query.

Similar as κ-NDCG, we can extend ERR to κ-ERR with the following definition.

$$\kappa - ERR = \sum_{s=1}^{n} \frac{1}{n_i} R(y_s^{(i)}) \prod_{t=1}^{s-1} (1 - R(y_t^{(i)}),\ R(r) = \frac{2^r - 1}{2^{y_{max}^{(i)}}}, \quad (3)$$

where $y_{max}^{(i)}$ is the relevance label in the top position, as defined in Section 3.2.1.

4. EXPERIMENTAL RESULTS

In this section, we empirically evaluate our proposed top-k labeling strategy and ranking model. Firstly, we conducted user studies to compare the effectiveness and efficiency of our top-k labeling strategy with traditional absolute judgment based on the dataset from the Topic Distillation task of TREC2003. Secondly, we compared FocusedRank with its corresponding traditional ranking models and top-k ListMLE [30] based on both the ground-truths from the absolute judgment and the top-k ground-truths. The experimental results show the superiority of our labeling strategy and ranking model to previous work.

[3]Since MAP is mainly designed for binary judgment scenario, we omit the modification on it.

Table 1: Comparison results of time efficiency

Method	Time per judgment(s)	Time per query(min)	Judgment complexity	#Judgments per query
Top-k labeling	5.51	13.13	$\mathcal{O}(n \log k)$	142.76
Five-grade judgment	13.87	11.78	$\mathcal{O}(n)$	50

4.1 Top-k Labeling

To study whether top-k labeling is "easier" to make than absolute relevance judgments, we compare the Top-k labeling strategy (i.e., $k = 10$) with the popular five-graded ("bad", "fair", "good", "Excellent", "perfect") absolute relevance judgment method. We investigate which one is a better judgment method under two basic metrics, namely time efficiency [26], and agreement among assessors [7].

4.1.1 Experiment Design

We describe our experimental design from the following four aspects:

Data Set: We adopted all the 50 queries from the Topic Distillation task of TREC2003 as our query set. For each query, we then randomly sampled 50 documents from its associated documents for judgment. Existing Web pages in corpus of TREC2003 were employed in labeling to avoid the time delay in downloading content from Internet. In TREC topics, most of the queries have clear intent in the form of query descriptions, which are also exhibited along with the queries in labeling for better understanding.

Labeling Tools: We designed labeling tools for two judgment methods separately, i.e., the top-k labeling tool $T1$ and the traditional five-graded relevance judgment tool $T2$. In $T1$, a query with its description is shown at the top, and two associated Web pages are placed at the main area. An assessor is then asked to decide which one is more relevant. In $T2$, a query with its description is shown at the top followed with five grade buttons, and a Web page is placed at the main area. An assessor is asked to decide which grade should be assigned to that page. A timer is introduced to both tools for computing time per judgment. Assessors can click the clock button to stop the timer if they want to have a break or leave for a while. This will ensure the computing accuracy.

Assessors: There are five assessors participating our user study. These assessors are all graduate students who are familiar with Web search. They all received a training in advance on how to use the tools and on the specifications of the five grades.

Assignment: To make the comparison valid, the assignment should meet the following requirements. Firstly, for each method, all the selected documents should be judged at least once to obtain a complete data set. Secondly, for each assessor, he/she is most likely to memorize some information on the documents for a given query after judging. Therefore, to compare the methods independently, we shall ensure that each assessor will not see the same query under different tools to minimize the possible order effect. Finally, each tool has to be utilized by all the assessors to avoid the possible differences between individuals. Therefore, we adopted the following assignment to satisfy all the requirements: (1) all the 50 queries are divided into five folds $\{Q_i\}_{i=1}^5$, where each fold has 10 queries; (2) for $i = 1, \ldots, 4$, each assessor U_i judges Q_i with $T1$ and Q_{i+1} with $T2$, and the assessor U_5 judges Q_5 with $T1$ and Q_1 with $T2$. At least such two different assignments are needed to compute the agreement among assessors.

	A$\succ$B	A$\sim$B	A$\prec$B
A$\succ$B	0.6749	0.2766	0.0485
A$\sim$B	0.1138	0.8198	0.0664
A$\prec$B	0.1047	0.3779	0.5174

Table 2: Assessor agreement for preference judgments in top-k labeling results

	A$\succ$B	A$\sim$B	A$\prec$B
A$\succ$B	0.6272	0.2913	0.0815
A$\sim$B	0.2825	0.5232	0.1944
A$\prec$B	0.1534	0.3826	0.4640

Table 3: Assessor agreement for inferred preference judgments in five-graded labeling results

4.1.2 Evaluation Results and Discussions

Two basic metrics are utilized to evaluate the labeling strategies.

(1) Time efficiency: Time efficiency [26] is used to measure the cost of labeling. It is dependent on the time per judgment and the complexity of judgment (the total number of judgments for each query). Statistically, less time per judgment suggests easier judgment.

(2) Agreement: Here we measure the agreement between two assessors over all judgments as [7] to average out differences in expertise, prior knowledge, or understanding of the query. For comparison, we inferred preferences from the absolute judgements: if the judgment on item A was greater than the judgment on item B, we inferred that A was preferred to B (denoted by A$\succ$B). A tie between A and B is denoted by A$\sim$B.

As shown in Table 1, it is obvious that the average time per judgment in absolute judgments is longer than that of the preference judgments in top-k labeling strategy, e.g. about $2 \sim 3$ times. The results verifies the common sense that the preference judgment is easier than absolute judgment. Meanwhile, from the average number of judgments conducted on each query, we can find that the top-k labeling strategy will take more judgments than the absolute judgments, at a scale around the theoretical value $\log k$ (i.e. k=10). Most importantly, we can see that the total judgment time spent on each query is comparable between the two methods. The results indicate that by adopting the top-k labeling strategy, the complexity of pairwise preference judgment becomes similar to that of the absolute judgment. Therefore, it is feasible to use top-k labeling in practice.

The agreement among assessors for preference judgment

Graded MQ2007 (Three-grade relevance judgments)

Methods	N@1	N@2	N@3	N@4	N@5	N@6	N@7	N@8	N@9	N@10	ERR
SVM^{MAP}	0.4006	0.4039	0.4111	0.4128	0.4165	0.4217	0.4266	0.4322	0.4363	0.4419	0.3146
RankSVM	**0.4118**	0.4058	0.4051	0.4099	0.4173	0.4216	0.4267	0.4320	0.4380	0.4447	0.3178
FocusedSVM	0.4060	0.4083	0.4077	0.4079	0.4123	0.4179	0.4234	0.4299	0.4351	0.4400	0.3196
AdaRank	0.3789	0.3941	0.3955	0.4024	0.4066	0.4119	0.4169	0.4225	0.4287	0.4345	0.3061
RankBoost	0.3972	0.4051	0.4062	0.4095	0.4150	0.4197	0.4260	0.4316	0.4374	0.4438	0.3101
FocusedBoost	0.3947	0.4098	0.4110	0.4132	0.4164	0.4235	0.4282	0.4317	0.4368	0.4422	0.3199
ListNet	0.4114	**0.4110**	**0.4134**	**0.4153**	**0.4204**	0.4243	0.4300	0.4332	0.4389	0.4442	0.3206
RankNet	0.4013	0.4086	0.4076	0.4118	0.4156	0.4211	0.4267	0.4330	0.4379	0.4451	0.3157
FocusedNet	0.3968	0.4103	0.4126	0.4153	0.4191	0.4245	**0.4301**	0.4341	**0.4403**	**0.4459**	**0.3223**
Top-k ListMLE	0.4057	0.4091	0.4115	0.4143	0.4188	**0.4247**	0.4293	**0.4346**	0.4392	0.4443	0.3168

Top-k MQ2007 (Top-k judgments)

Methods	κ-N@1	κ-N@2	κ-N@3	κ-N@4	κ-N@5	κ-N@6	κ-N@7	κ-N@8	κ-N@9	κ-N@10	κ-ERR
SVM^{MAP}	0.4530	0.5023	0.5389	0.5702	0.5951	0.6178	0.6346	0.6479	0.6591	0.6690	0.6227
RankSVM	0.4545	0.4932	0.5315	0.5664	0.5930	0.6135	0.6306	0.6445	0.6564	0.6655	0.6205
FocusedSVM	0.4539	0.5087	0.5448	0.5773	0.6024	0.6224	0.6404	0.6527	**0.6646**	**0.6739**	0.6287
AdaRank	0.3896	0.4438	0.4745	0.5062	0.5358	0.5575	0.5777	0.5952	0.6089	0.6190	0.5637
RankBoost	0.4438	0.4825	0.5243	0.5567	0.5850	0.6066	0.6234	0.6361	0.6476	0.6571	0.6131
FocusedBoost	0.4529	0.4943	0.5312	0.5633	0.5918	0.6123	0.6288	0.6420	0.6523	0.6628	0.6187
ListNet	0.4541	0.4937	0.5314	0.5669	0.5922	0.6132	0.6299	0.6419	0.6530	0.6613	0.6195
RankNet	0.4490	0.4875	0.5269	0.5586	0.5865	0.6077	0.6249	0.6390	0.6512	0.6603	0.6143
FocusedNet	**0.4623**	**0.5108**	**0.5460**	**0.5783**	**0.6028**	**0.6240**	**0.6409**	**0.6529**	0.6633	0.6735	**0.6336**
Top-k ListMLE	0.4570	0.4991	0.5356	0.5719	0.5969	0.6176	0.6339	0.6476	0.6585	0.6673	0.6228

Table 4: Performance comparison on Graded MQ2007 and Top-k MQ2007

in top-k labeling and for inferred preference judgment in absolute judgment is shown in Table 2 and Table 3, respectively. Each cell (X_1,X_2) is the probability that one assessor would say X_2(column) given that another assessor said X_1(row). Therefore, they are row normalized. From the results, we can see that by adopting preference judgment, top-k labeling can largely improve the agreement among assessors over the absolute judgment. The overall agreement among assessors reaches 74.5% under top-k labeling, while it is only 54.7% under absolute judgment. We conducted χ^2 test to compare the ratio of the number of pairs agreed on to the number disagreed on for both top-k labeling and absolute judgment. The difference is significant ($\chi^2 = 420.7, df = 1, p < 0.001$). We also investigate the agreement among assessors only on preference pairs (by ignoring ties), where the agreement ratio is 89.7% under top-k labeling, and 83.1% under absolute judgment. We test the difference between the two methods using the ratio of agreed preference pairs to disagreed preference pairs, which is also significant ($\chi^2 = 28.5, df = 1, p < 0.001$).

From the above results, we can conclude that the top-k labeling strategy is both efficient and effective to obtain reliable judgments from human assessors, as compared with traditional absolute judgment.

4.2 Performance of FocusedRank

In this section, we empirically evaluate the performance of our proposed ranking model, i.e. FocusedRank. Specially, we conducted extensive experiments to compare FocusedRank with different state-of-the-art ranking models based on both the ground-truths from the absolute judgment and the top-k ground-truths. Note that k is set to 10 in our experiments. Besides, we also investigated the impact of the balance factor β in our proposed FocusedRank.

4.2.1 Experimental Settings

For comparison, we constructed two datasets, each with both the absolute judgment and top-k labeling. One dataset comes from the benchmark LETOR4.0 collection. There are two homologous datasets with different labeling in LETOR4.0, one is referred to as MQ2007 with three-graded relevance judgments, and the other is MQ2007-list with the total order judgments as the ground-truth. The two datasets share the same queries. The only difference lies in that the documents of a query in MQ2007 is the subset of the documents of the corresponding query in MQ2007-list. Thus the intersection of two document sets on each query is adopted. Those from MQ2007 comprise the ground-truth with absolute judgments, referred as Graded MQ2007. While those from MQ2007-list become the top-k ground-truth by only preserving the total order of top k documents on each query, referred as top-k MQ2007.

The other dataset is the one manually constructed in previous user study experiments with 50 queries from the TREC-2003 Topic Distillation task. The one with the five-graded absolute relevance judgments is denoted as Graded TD2003, and the one with top-k labeling is denoted as Top-k TD2003.

We divided each dataset into five subsets, and conducted 5-fold cross-validation. In each trial, three folds were used for training, one fold for validation, and one fold for testing. For RankSVM and SVM^{MAP} the validation set in each trial was used to tune the coefficient C. For RankNet and ListNet it was used to determine the number of iterations. For our FocusedRank, the validation set was used to tune the balance factor β.

Graded TD2003 (Five-grade relevance judgments)

Methods	N@1	N@2	N@3	N@4	N@5	N@6	N@7	N@8	N@9	N@10	ERR
SVM^{MAP}	0.5055	0.5356	0.5335	0.5374	0.5515	0.5584	0.5669	0.5743	0.5763	0.5801	0.5102
RankSVM	0.5019	0.5409	0.5446	0.5657	**0.5816**	**0.5829**	0.5867	0.5847	0.5905	0.5991	0.5072
FocusedSVM	0.5126	0.5505	0.5498	0.5633	0.5642	0.5656	0.5642	0.5761	0.5840	0.5885	0.5129
AdaRank	0.5278	0.5436	0.5619	**0.5740**	0.5774	0.5797	0.5845	0.5872	0.5944	0.5982	0.5132
RankBoost	0.5230	0.5094	0.5179	0.5362	0.5456	0.5507	0.5555	0.5623	0.5640	0.5750	0.5119
FocusedBoost	**0.5486**	0.5275	0.5356	0.5532	0.5554	0.5706	0.5790	0.5856	0.5876	0.5955	0.5466
ListNet	0.5229	0.5480	0.5514	0.5663	0.5736	0.5804	**0.5885**	**0.5949**	0.5944	0.5985	0.5225
RankNet	0.4971	0.5304	0.5612	0.5638	0.5642	0.5765	0.5817	0.5837	0.5895	0.5983	0.5059
FocusedNet	0.5265	**0.5660**	**0.5642**	0.5706	0.5780	0.5804	0.5848	0.5898	**0.5948**	**0.6082**	**0.5660**
Top-k ListMLE	0.4994	0.5190	0.5356	0.5504	0.5540	0.5717	0.5746	0.5798	0.5861	0.5885	0.5083

Top-k TD2003 (Top-k judgments)

Methods	κ-N@1	κ-N@2	κ-N@3	κ-N@4	κ-N@5	κ-N@6	κ-N@7	κ-N@8	κ-N@9	κ-N@10	κ-ERR
SVM^{MAP}	0.2248	0.2822	0.2884	0.2973	0.3271	0.3359	0.3459	0.3606	0.3769	0.3858	0.3839
RankSVM	0.2246	0.2893	0.2921	0.3083	0.3253	0.3387	0.3561	0.3703	0.3794	0.3872	0.4025
FocusedSVM	0.2243	0.2965	0.2958	0.3192	0.3235	0.3415	0.3662	**0.3801**	0.3819	0.3886	0.4041
AdaRank	0.2029	0.2979	0.2995	0.3116	0.3289	0.3379	0.3427	0.3574	0.3656	0.3766	0.3777
RankBoost	0.2082	0.2931	0.2921	0.3103	0.3255	0.3365	0.3576	0.3654	0.3782	0.3789	0.3970
FocusedBoost	0.2583	0.3168	0.3124	0.3191	0.3346	0.3517	0.3634	0.3732	0.3861	0.3980	0.4214
ListNet	0.2685	0.2731	0.2694	0.2789	0.3044	0.3142	0.3252	0.3404	0.3514	0.3624	0.3962
RankNet	**0.2776**	0.2946	0.2946	0.3021	0.3227	0.3372	0.3470	0.3688	0.3763	0.3813	0.4199
FocusedNet	0.2473	**0.3268**	**0.3237**	**0.3223**	**0.3507**	**0.3605**	**0.3705**	0.3769	0.3849	**0.4058**	**0.4603**
Top-k ListMLE	0.2389	0.2861	0.3055	0.3222	0.3480	0.3552	0.3668	0.3702	**0.3913**	0.4007	0.4028

Table 5: Performance comparison on Graded TD2003 and Top-k TD2003

Besides, in our experiments, when applying traditional ranking models on top-k ground-truths, we relate the top-k ground-truth to absolute labels as defined in Section 3.2.1. While applying top-k ranking models (i.e. FocusedRank and top-k ListMLE) on absolute judgment datasets, we randomly generate a total order of the documents according to graded labels and preserve the top-k order for learning.

To measure the effectiveness of ranking performance, NDCG [13] and ERR [8] are used on ground-truths from absolute judgments, while κ-NDCG and κ-ERR are adopted on top-k ground-truths.

4.2.2 Comparison results

The performance comparison between different ranking models on the two datasets is shown in Table 4 and Table 5, respectively.

From the results on the Graded MQ2007 as shown in the upper part of Table 4, we can see that the overall performance of FocusedRank is comparable with traditional pairwise and listwise ranking models in terms of both NDCG ($N@j$) and ERR. It shows that even though FocusedRank is proposed for the top-k ground-truth, it can work quite well on traditional absolute judgment datasets under traditional IR measures. Such results also reveals that, learning the ordering of the top items well is critical for the success of a learning to rank algorithm. Similar results can also been found on the Graded TD2003 datasets as shown in the upper part of Table 5.

From the results on top-k ground-truths in both tables (i.e. the bottom parts), we can find that FocusedRank can significantly outperform the corresponding pairwise and listwise ranking models in terms of both κ-NDCG (κ-$N@j$) and κ-ERR. For example, considering FocusedBoost, the relative improvement over AdaRank and RankBoost is about 7.08% and 0.87% in terms of κ-NDCG@10, respectively, and the relative improvements in terms of κ-ERR is about 9.76% and 0.91%, respectively. Besides, we can also observe that under all the metrics, the best performance is almost reached by our FocusedRank (The best performance is denoted by number in bold). The results indicate that FocusedRank is particularly suitable for the top-k ground-truth. By combining the advantages of both pairwise and listwise approaches, FocusedRank can fully exploit the information in the top-k ground-truth and thus outperforms each single model.

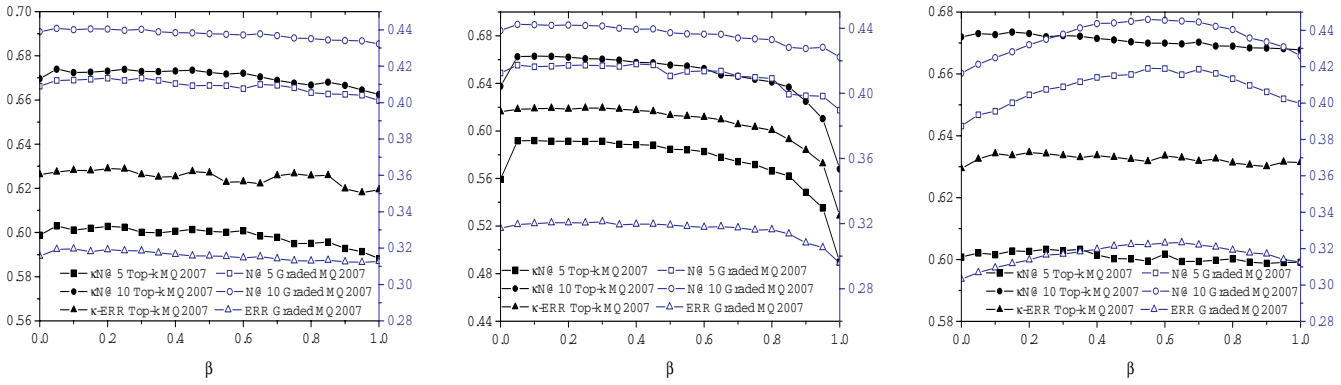
Moreover, when we compare FocusedRank with the state-of-the-art top-k ranking model, i.e., top-k ListMLE, we can see comparable performances on both absolute judgment datasets and top-k ground-truths. In fact, some of our FocusedRank model, e.g. FocusedNet, can consistently outperform top-k ListMLE on Top-k MQ2007 in terms of both κ-NDCG and κ-ERR. The results demonstrate that FocusedRank, as a mixed ranking model, can effectively cope with the top-k learning to rank problem.

4.2.3 The impact of the balance factor β

Here we investigate the impact of the balance factor β in FocusedRank. By varying β from 0 to 1 with a step of 0.05, the curves of the ranking performance of FocusedRank in terms of κ-NDCG[4] and κ-ERR are shown in Figure 1 and Figure 2. Each performance value on test sets shown in the figures is averaged using five-fold cross validation as the same way used in LETOR.

In Figure 1, the performance variations on graded MQ2007 are represented as curves with open symbols while that on

[4]For space limitation, we just show the results of @5,@10 for NDCG and κ-NDCG.



(a) FocusedSVM (b) FocusedBoost (c) FocusedNet

Figure 1: Performance variation of FocusedRank with β on Graded MQ2007 and Top-k of MQ2007 under corresponding evaluation measures.

Top-k MQ2007 are represented as curves with filled symbols. The results of FocusedSVM, FocusedBoost and FocusedNet are shown in Figure 1(a), (b) and (c), respectively. To make the variation trend more clear, each figure adopts double y-axes, where the black curves of κ-NDCG@5 (κ-N@5), κ-NDCG@10 (κ-N@10) and κ-ERR use the left y axis, while the other blue curves of NDCG@5 (N@5), NDCG@10 (N@10) and ERR utilize the right y axis. Similarly, in Figure 2 we also use double y-axes. The performance variations on graded TD2003 are represented as curves with open symbols while that on Top-k TD2003 are represented as curves with filled symbols. The results of FocusedSVM, FocusedBoost and FocusedNet are shown in Figure 2(a), (b) and (c), respectively.

From the results shown in Figure 1 and Figure 2, we find that: (1) There is a consistent trend[5] for the three types of FocusedRank on the two groups of different datasets. That is, as β increases, the ranking performance first grows to reach its maximum, and then drops. (2) The overall variation of each performance curve is small. Take the most varied curves for example, the variance of the mean is 2.7% for the performance curve of κ-NDCG@5 on Top-k MQ2007, and the variances of the mean is 6.2% for that on Top-k TD2003.

Thus, we come to a conclusion that the performance of FocusedRank is relative stable with respect to β.

5. CONCLUSIONS

In this paper, we propose a novel top-k learning to rank framework, including labeling, ranking and evaluation, which can be effectively adopted for real search systems. Firstly, a top-k labeling strategy is proposed to obtain reliable relevance judgments from human assessors via pairwise preference judgment. With this labeling strategy, we can largely reduce the complexity of pairwise preference judgment to $\mathcal{O}(n \log k)$. Secondly, a novel ranking model FocusedRank is presented to capture the characteristics of the top-k ground-truth. Thirdly, two new top-k evaluation measures are derived to fit the top-k ground-truth. We verify the efficiency and reliability of the proposed top-k labeling strategy through user studies, and demonstrate the effectiveness of top-k ranking model by comparing with state-of-the-art ranking models.

There are many interesting issues for further investigation under our top-k learning to rank framework. (1) The top-k labeling strategy could be improved to further reduce the judgment complexity. For example, we may introduce the "Bad" judgment like [7] for pages that are clearly irrelevant to further save labeling effort. (2) With top-k ground-truth, the design for new ranking models remains a valuable problem to investigate. (3) It is also possible to find new top-k based evaluation measures for better comparison between different systems.

[5]The small size of the datasets may be the reason for the non-smooth variation curves with 50 queries on Graded TD2003 and Top-k TD2003, compared with more than one thousand queries on Graded MQ2007 and Top-k MQ2007.

6. ACKNOWLEDGMENTS

This research work was funded by the National Natural Science Foundation of China under Grant No. 60933005, No. 61173008, No. 61003166 and 973 Program of China under Grants No. 2012CB316303.

7. REFERENCES

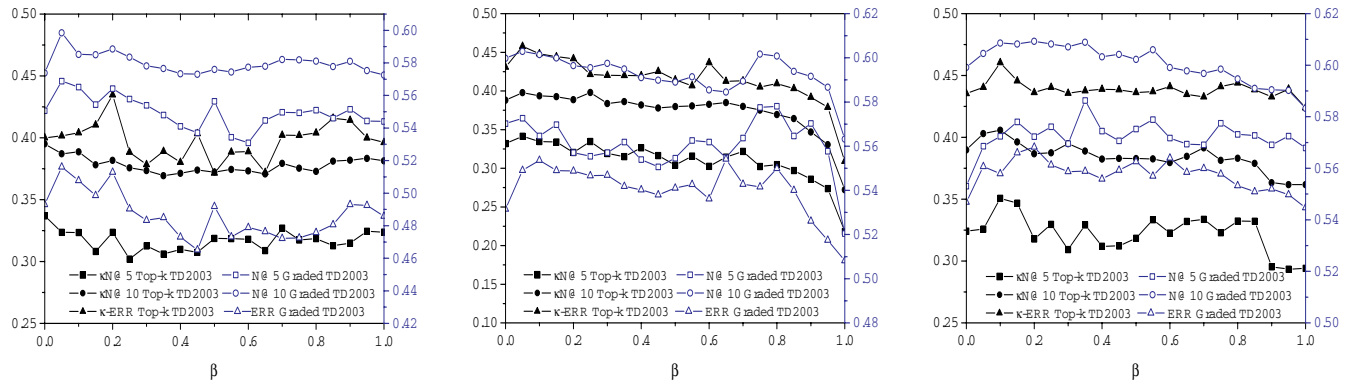
[1] N. Ailon and M. Mohri. An efficient reduction of ranking to classification. COLT '08, pages 87–98, 2008.

[2] C. Buckley and E. M. Voorhees. *Retrieval system evaluation*, chapter TREC: experiment and evaluation in information retrieval. MIT press, 2005.

[3] C. Burges, T. Shaked, and et al. Learning to rank using gradient descent. ICML '05, pages 89–96, 2005.

[4] R. Burgin. Variations in relevance judgments and the evaluation of retrieval performance. *IPM*, 28:619–627, 1992.

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



(a) FocusedSVM (b) FocusedBoost (c) FocusedNet

Figure 2: Performance variation of FocusedRank with β on Graded TD2003 and Top-k TD2003 under corresponding evaluation measures.

[6] B. Carterette and P. N. Bennett. Evaluation measures for preference judgments. SIGIR '08, pages 685–686, 2008.

[7] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais. Here or there: preference judgments for relevance. ECIR'08, pages 16–27, 2008.

[8] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. CIKM '09, pages 621–630. ACM, 2009.

[9] S. Clémençon and N. Vayatis. Ranking the best instances. *JMLR*, 8:2671–2699, 2007.

[10] D. Cossock and T. Zhang. Subset ranking using regression. *Learning theory*, 4005:605–619, 2006.

[11] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *JMLR*, 4:933–969, 2003.

[12] E. P. C. III. The optimal number of response alternatives for a scale: A review. *Journal of Marketing Research*, 17, No. 4:407–422.

[13] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. SIGIR '00, pages 41–48, 2000.

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

[15] J. Kekäläinen. Binary and graded relevance in ir evaluations-comparison of the effects on ranking of ir systems. *IPM*, 41:1019–1033, 2005.

[16] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Syst.*, 27:2:1–2:27, 2008.

[17] L. P., B. C., and W. Q. Mcrank: learning to rank using multiple classification and gradient boosting. In *NIPS2007*, pages 845–852.

[18] T. Qin, T.-Y. Liu, and et al. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13:346–374, 2010.

[19] T. R., S. W. Jr., and V. J.L. Towards the identification of the optimal number of relevance categories. *JASIS*, 50:254–264, 1999.

[20] K. Radinsky and N. Ailon. Ranking from pairs and triplets: information quality, evaluation methods and query complexity. WSDM '11, pages 105–114, 2011.

[21] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. KDD '05, pages 239–248, 2005.

[22] F. G. Rebecca and N. Melisa. The neutral point on a likert scale. *Journal of Psychology*, 95:199–204, 1971.

[23] P. R.L. The analysis of permutations. *Applied Statistics*, 24(2):193–202, 1974.

[24] M. Rorvig. The simple scalability of documents. *JASIS*, 41:590–598, 1990.

[25] C. Rudin. Ranking with a p-norm push. In *COLT*, pages 589–604, 2006.

[26] R. Song, Q. Guo, R. Zhang, and et al. Select-the-best-ones: A new way to judge relative relevance. *IPM*, 47:37–52, 2011.

[27] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. SIGIR '98, pages 315–323. ACM, 1998.

[28] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *IPM*, 36:697–716, 2000.

[29] E. M. Voorhees. Evaluation by highly relevant documents. SIGIR '01, pages 74–82. ACM, 2001.

[30] F. Xia, T.-Y. Liu, and H. Li. Statistical consistency of top-k ranking. In *NIPS*, pages 2098–2106, 2009.

[31] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. SIGIR '07, pages 391–398, 2007.

[32] Yao. Measuring retrieval effectiveness based on user preference of documents. *JASIS*, 46:133–145, 1995.

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

[34] B. Zhou and Y. Yao. Evaluating information retrieval system performance based on user preference. *JIIS*, 34:227–248, 2010.