7-2011

Relevant knowledge helps in choosing right teacher: Active query selection for ranking adaptation

Peng CAI

Wei GAO
Singapore Management University, weigao@smu.edu.sg

Kam-Fai WONG

Aoying ZHOU

Relevant Knowledge Helps in Choosing Right Teacher: Active Query Selection for Ranking Adaptation

Peng Cai[1], Wei Gao[2], Aoying Zhou[1], Kam-Fai Wong[2,3]

[1]East China Normal University, Shanghai, China
pengcai2010@gmail.com, ayzhou@sei.ecnu.edu.cn
[2]The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
{wgao, kfwong}@se.cuhk.edu.hk
[3]Key Laboratory of High Confidence Software Technologies, Ministry of Education, China

ABSTRACT

Learning to adapt in a new setting is a common challenge to our knowledge and capability. New life would be easier if we actively pursued supervision from the right mentor chosen with our relevant but limited prior knowledge. This variant principle of active learning seems intuitively useful to many domain adaptation problems. In this paper, we substantiate its power for advancing automatic ranking adaptation, which is important in web search since it's prohibitive to gather enough labeled data for every search domain for fully training domain-specific rankers. For the cost-effectiveness, it is expected that only those most informative instances in target domain are collected to annotate while we can still utilize the abundant ranking knowledge in source domain. We propose a unified ranking framework to mutually reinforce the active selection of informative target-domain queries and the appropriate weighting of source training data as related prior knowledge. We select to annotate those target queries whose documents' order most disagrees among the members of a committee built on the mixture of source training data and the already selected target data. Then the replenished labeled set is used to adjust the importance of source queries for enhancing their rank transfer. This procedure iterates until labeling budget exhausts. Based on LETOR3.0 and Yahoo! Learning to Rank Challenge data sets, our approach significantly outperforms the random query annotation commonly used in ranking adaptation and the active rank learner on target-domain data only.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

SIGIR'11, July 24–28, 2011, Beijing, China.

Keywords

Ranking adaptation, active learning, query by committee

1. INTRODUCTION

In recent years, a lot of learning to rank (or rank learning) algorithms have been proposed for information retrieval (IR) as an effective relevance ranking approach to ordering search results [27]. Other than document ranking, learning to rank was also widely applied to many real-world domains of application such as online computational advertisement [26], recommendation [41], key phrase extraction [23] and question and answering [37].

General rank learning algorithms follow the supervised learning paradigm which requires large amount of training data that are usually expensive and time-consuming to obtain. This limit is striking in particular for ranking because relevance judgment is query dependent in terms of multiple ratings, requiring annotators to carefully assess not only the similarity between each retrieved document and the query but also the preference order among different documents. Giving a new ranking field, therefore, it is often desperately desired that somehow we can keep the work of annotation minimum while maximizing the effectiveness of learning at same time. Such kind of effort is especially beneficial to the quick deployment of ranking model into different search domains for applications such as vertical search and cross-language search towards different markets.

Domain adaptation and active learning are two general learning paradigms that are proposed to save labeling costs from fundamentally different perspectives. Each of them has been successfully applied in rank learning. Ranking adaptation [8, 9, 10, 11, 17, 18, 19, 40] focuses on the exploration of cross-domain relatedness of ranking knowledge by reusing the training data in large amount from another domain (i.e. source domain) to help the rank learning in the designated domain (i.e. target domain) where no or just a few labeled data is available. Active rank learning [2, 15, 16, 28, 45] is motivated to proactively select and annotate those most informative set of examples, which are expected that, if labeled, they can maximize the information value to the ranking function.

Despite of their different motivations and mechanisms, the two approaches seem to be inherently complementary in a sense that the deficiency of one method can be naturally remedied by the strengths of another. Active rank learn-

ing is characterized as the proactive seeking for feedback from domain expert (e.g. an oracle), and it prevents from using prior knowledge that sometimes is abundantly in existence elsewhere. In contrast, ranking adaptation advocates the cross-domain transfer of prior ranking knowledge from a related domain, which is controlled by resorting to merely a small random set of labeled examples in target domain, not allowing for any feedback for receiving more informative supervision. Therefore, we may take advantage of both approaches for the cost-effective rank learning.

There is little research on combining the two paradigms even in the general machine learning study. For this purpose, we need to conquer two obvious challenges: (1) how to effectively bridge the shift of data across domains in terms of different joint distributions of rank features and rank labels; (2) how to select the minimum set of target-domain examples to label that can well represent the ranking information of the target domain. In this work, we propose to combine the utilization of relevant knowledge in source-domain training data and the active selection of informative queries from target domain, where the performance of these two components can be mutually reinforced within a unified iterative framework. Specifically, we first select those target-domain queries whose document order most disagree among the members of a committee, which is built on the mixture of source-domain training data and the already selected target data. Then the replenished labeled set is used to adjust the importance weights of source-domain queries for boosting the transfer of prior ranking knowledge. The two components proceed alternately until the labeling budget exhausts. With the experiments on LETOR3.0 and Yahoo! Learning to Rank Challenge data sets, the results show that our method outperforms two strong baselines: one is based on the random query annotation commonly used in ranking adaptation and the other is the active rank learning on target-domain data only. Note that our method, although discussed in particular for ranking problem herein, can be generalized to other learning scenarios.

The rest of the paper is organized as follows: We review the related work in Section 2; Section 3 introduces the concept of informative knowledge for adaptation; Section 4 describes active rank learning algorithm using query-by-committee; Section 5 presents the unified learning framework of active query selection for ranking adaptation; Section 6 discusses experiments and results; Finally, we conclude in Section 7.

2. RELATED WORK

In this section, we outline the literature related to ranking adaptation and active rank learning. The fundamentals of learning to rank can be found in the comprehensive survey [27], which are not reviewed here due to space limit.

2.1 Ranking Adaptation

In essence, domain adaptation deals with the learning setting where the training and test examples are drawn from different distributions, which is referred to as dataset shift problem [31]. It is hypothesized that common information between two domains can be found and used to bridge the shift across domains in learning. Most domain adaptation studies are focused on classifier adaptation [3, 13, 14, 22, 46]. The main concentration of rank learning is the preference order or the full order of multiple documents instead of absolute class label. It is thus not straightforward to directly apply classifier adaptation for ranking. Therefore, ranking adaptation was received more and more attention.

Usually we have large amount of source-domain training examples, but have only a small set of training examples in target domain. Various ranking adaptation algorithms have been proposed under this setting [8, 9, 10, 11, 17, 18, 19, 40]. In [11, 19], the parameters of ranking model trained on source-domain data was adjusted with the target-domain labeled data. [9, 10] presented instance-based and feature-based adaptation approaches with the help of a few target-domain training data. [40] focused on ranking adaptation among heterogeneous domains. [17] learned ranking models on two domains separately and then constructed a stronger model by interpolating them. [18] proposed a ranking adaptation framework based on instance weighting in a more relaxed setting where target-domain data need not be labeled. [8] modeled a multi-task learning algorithm based on boosted decision trees, where the commonalities among different ranking tasks were leveraged to enhance those specific tasks with only few training examples.

Existing methods assume that the small set of target-domain training examples carry domain-specific information that can be used to guide the identification of relevant examples in source domain and the transfer of their ranking knowledge to target domain. However, this is often invalid since the target training set is commonly very small and is predetermined randomly rather than purposefully. Therefore, it can hardly well characterize the target domain. In this work, we intend to find those most specific target-domain examples to annotate for ranking adaptation.

2.2 Active Rank Learning

The motivation of active learning is to put limited human resource on annotating most informative examples. Many strategies were proposed to measure the informativeness of unlabeled examples. The simplest one is uncertainty sampling [24], where the examples whose predicted label is the most uncertain are deemed informative. Another typical strategy is the query-by-committee (QBC) algorithm [34], where a committee is formed by a set of *diverse* hypotheses trained on currently labeled data. The informative examples are the ones whose labels most disagree among committee members. Expected model change is yet another active learning framework where an example is labeled if knowing its label tends to incur some significant change on the current model. Similarly, expected error reduction assesses the expected generalization error reduced if the label is obtained for an example. The comprehensive survey of active learning is given in [33].

Active learning has been actively extended to rank learning [2, 15, 16, 28, 45]. Based on uncertainty sampling, [45] selected the most ambiguous document pairs, in which two documents received close scores predicted by the current model, as informative examples. [15] chose those document pairs, which if labeled could change the current model parameters significantly. [16] addressed active rank learning based on expected hinge rank loss minimization criterion. Inspired by expected loss reduction strategy, [28] recently introduced an expected loss optimization framework for ranking, where the selection of query and documents were integrated in a principled manner.

A closely related problem to active rank learning is the

active feedback for ad-hoc information retrieval [35, 42]. [35] presented a general framework for active feedback based on diversity-based selection algorithm, which outperformed traditional relevance feedback that simply selects the top K documents. [42] described a Bayesian logistic regression model for active feedback using the variance reduction approach to capture relevance, diversity and uncertainty of the unlabeled documents.

2.3 The Hybrid Approach

The combination of domain adaptation and active learning is potentially more powerful and cost-effective, but so far has not been well studied. In natural language processing, [6] used naive Bayes classifier with active leaning based on uncertainty sampling for word sense disambiguation in a domain adaptation setting. The selection was made in the entire set of target-domain instances. For sentiment classification, [32] leveraged the source-domain information to learn a best initial hypothesis for active selection and ruled out target instances on the source side using the domain separator hypothesis. [36] presented an active transfer learner, where a few labeled examples must be provided initially for training a target-domain classifier which is the basis of active knowledge transfer. In cross-domain video concept detection, [25] proposed a hybrid selection strategy by combining discriminative strategy that selected target instances near the decision boundary and generative strategy that chose target samples unlikely generated by the source distribution. The intent of the two strategies was to deal with the different extent of distribution divergence between the domains. To our best knowledge, there is no hybrid approach studied for rank learning.

3. INFORMATIVE KNOWLEDGE TO SELECT FOR ADAPTATION

Domain adaptation aims to use source-domain training data as prior knowledge to help construct a model for target domain tasks by leveraging the common information to bridge the gap between two domains. Figure 1 conceptually illustrates the high-level data distribution in two related domains. As we can understand, the instances in between (in the blue ellipse) may contain some general cross-domain knowledge and the other two groups of instances fallen apart (in the green and red ellipses) only encode the specific knowledge of their own domains, thus referred to as cross-domain instances and domain-specific instances, respectively.

Note that the source-domain instances (red circles and rectangles) are assumed as all labeled, in which those cross-domain instances (red rectangles) tend to be more useful than domain-specific instances (red circles) for the training of adaptation model. Meanwhile, to annotate those cross-domain instances in target domain (green rectangles) is largely not required because such equivalent knowledge can be obtained from the cross-domain instances in source domain. As a result, no matter there is labeled instances in target domain or not, the most critical issue is how to obtain the domain-specific knowledge of target domain (green circles). If no target-domain instances are labeled, it is reasonable to selectively label some of them for bridging the distribution gap between two domains. On the other hand, even if a small set of training data are available in target domain, it is still unknown whether they can provide important domain-specific knowledge because they are usually *randomly* sampled for annotation. Therefore, it is necessary to *purposefully* select and annotate those most informative instances in target domain for providing domain-specific knowledge which is not contained in the source-domain training data.

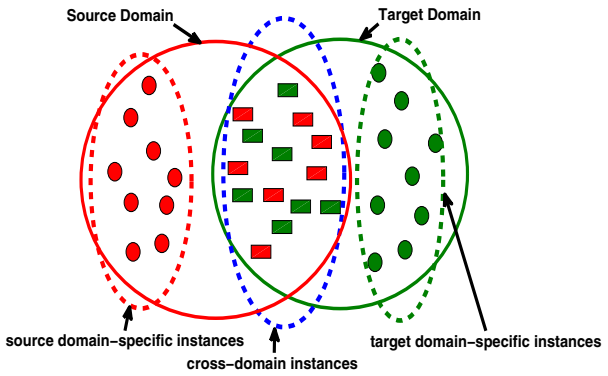


Figure 1: A conceptual illustration of data distribution of two domains. The cross-domain instances from source domain can help in learning the ranking model for target domain. Proactively selecting to label the target domain-specific instances is important to the success of adaptation.

One is likely to argue that with the initial set of labeled data in target domain people can simply perform in-domain active learning with no need of adaptation. We emphasize, however, that since the initial set is usually small and random, one must be dedicated to select wide spectrum of instances to strengthen the diversity of the labeled set. With the out-of-domain training data, it is expected that we need not label the cross-domain instances in target domain as the equivalent relevant knowledge can be acquired from the cross-domain instances in source domain, and thus can only focus on selecting and labeling those most specific target-domain instances for acquiring the informative knowledge useful for adaptation. Therefore, the overall labeling cost can be saved as compared to in-domain active learning.

4. QBC-BASED ACTIVE RANK LEARNING

Before elaborating our entire active rank learning framework in the next section, we discuss some essential issues on how pure active learning is applied to ranking. As compared to active classification, one of the most important differences is that in active rank learning it can be more flexible to choose informative examples at various levels such as query level [43, 44], document level [5, 15, 16, 45] or the combination of both [28]. Both query-level and document-level selection strategies have their own drawbacks. Query-level selection tends to waste labeling effort on non-informative documents when the number of documents for each query is large. Document-level selection assumes unrealistically that the documents or document pairs are independent of queries and may lead to the missing of informative queries or undesirable labeling results for rank learning [28]. For example, only one document is selected in a query or all the documents selected in a query have the same rating. Without

the loss of generality, we adopt query-level selection because of three reasons: (1) The generation of preference order patterns are prescribed within the scope of queries. The preference order of documents across queries are meaningless for web search ranking; (2) The informativeness of a selected query is more straightforward to measure with various rank evaluation criteria based on the list of its documents; (3) It is easy to extend query-level selection by further considering the informative documents associated with the query.

In this section, we present active query selection for ranking based on the typical query-by-committee (QBC) technique [34]. The QBC-based algorithm is implemented with two necessary components. The first is to build a committee of ranking models that are well diversified and compatible with the currently labeled data. The second is to measure the informativeness of queries by calculating the disagreement among the committee members on their ranking of search results, which aims at the selection of optimal queries. We describe the two components in Section 4.1 and 4.2.

4.1 Construct the Committee

A committee consists of multiple ranking models as committee members. Previously, different methods for building the committee have been proposed according to the genre of committee members. For generative model, the members can be obtained by randomly sampling the posterior distribution of model parameters [29]. For discriminative model, Abe and Mamitsuka [1] proposed two algorithms called query-by-boosting and query-by-bagging based on the ensemble leaning methods boosting and bagging [4]. We use query-by-bagging considering that most ranking algorithms are discriminative and bagging is a more general and flexible in this setting. Given the set of currently labeled instances D_l, bagging generates T partitions of sub-samples by sampling uniformly from D_l with replacement, and then the committee can be constructed by training each of its members on one portion of the sub-sample partitions.

Algorithm 1 details our QBC-based active selection algorithm using bagging. Suppose we initially have a small base set of queries and their labeled documents (*labeled queries* for short throughout the rest of paper). Similarly, we sample with replacement for T times in the set of labeled queries and train a ranking model on each subset of queries. In step 3, we set the size of each subset to be $m\%$ of $|D_l|$ and m is fixed so that the size of subsets keeps growing with the increase of $|D_l|$. In step 5, the informative query q_i is selected for annotation whose document ranking most disagrees among the T members (see Section 4.2 for the measurement on ranking disagreement). With the required number of informative queries being selected and annotated, the final ranking model H is trained on D_l in step 9.

4.2 Measure Ranking Disagreement

Given a query q, the committee members $h_1, h_2, \ldots, h_T$ return T ranked lists. We need to measure the inconsistency of these different rankings. The query whose documents are the most inconsistently ranked by the members is considered as informative query. All the traditional disagreement measures used by QBC are for classification purpose. Two well-known measures are vote entropy [12] and average KL-divergence [29]. For the KL-divergence-based metrics, the probability distribution of instance labels need be accurately estimated, and it cannot be directly extended to ranked list

Algorithm 1 Active rank learning based on bagging

Input:
N: The number of queries to be labeled
T: The size of committee
D_b: The base set of queries initially labeled, $|D_b| > 0$
D_l: Labeled queries, $|D_l| = 0$
D_u: Unlabeled queries, $|D_u| > 0$

Output:
Ranking model H derived from D_l;

1: $D_l \leftarrow D_b$;
2: **for** $i = 1$; $i \leq N$; $i++$ **do**
3: Sample queries in D_l uniformly with replacement, obtain subsets $S_1, S_2, \ldots, S_T$, each with size $|D_l| \times m\%$;
4: Train on each subset and obtain ranking models $h_1, h_2, \ldots, h_T$;
5: Find $q_i \in D_u$ whose ranking list most disagree among $h_1, h_2, \ldots, h_T$;
6: Obtain the rank label of each document in q_i;
7: $D_l \leftarrow D_l + q_i$, $D_u \leftarrow D_u - q_i$;
8: **end for**
9: Training H on D_l;
10: **return** H;

for the popular pairwise or listwise algorithms. Inspired by vote entropy, we find that the members can vote on each partial order of any document pair $\langle d_i \succ d_j \rangle$, where d_i has a higher ranking score than d_j. Given q and its corresponding ranked list D_q, the vote entropy of q can be defined as follows:

$$VE(q) = -\frac{1}{T} \sum_{d_i, d_j \in D_q} V(\langle d_i \succ d_j \rangle) \log \frac{V(\langle d_i \succ d_j \rangle)}{T} \quad (1)$$

where $V(\langle d_i \succ d_j \rangle)$ denotes the number of votes given by committee members who agree that d_i is ranked higher than d_j, and T is the size of committee. The query with maximum vote entropy will be selected from the unlabeled set for annotation. That is, $q^* = \text{argmax}_q \{VE(q)\}$.

5. ACTIVE RANKING ADAPTATION

Ranking adaptation requires finding the most informative target-domain knowledge as the right teacher for guiding the cross-domain transfer of ranking knowledge. As discussed in Section 3, the cross-domain instances from source domain shown in Figure 1 carry the common knowledge relevant to target domain. Such kind of common knowledge from source domain only need to be identified for reuse but not to be taught de novo since their rank labels already exist. Once identified, such kind of information can be used to choose those most domain-specific instances in target domain for annotation so that the transfer of ranking knowledge can take place. Clearly, two important problems need to be solved: (1) How to identify the cross-domain instances from source domain? (2) How could the identified cross-domain knowledge be exploited to choose target domain-specific instances for annotation?

We address the first problem by query weighting scheme where an appropriate importance or commonality measure is introduced to evaluate the degree of relevance of each source-domain query instance relative to the target domain. It can be expected that the cross-domain queries receive higher weights than domain-specific queries. Inspired by QBC, we

approach the second problem by selecting those informative queries from target domain with the help of cross-domain queries that are appropriately weighted. In return, with more target domain-specific queries labeled, it is expected to benefit the weighting of source queries in the first problem.

In this section, we model these two problems in a unified framework to mutually reinforce the source query weighting and the active selection of informative target queries by leveraging their reciprocal nature.

5.1 The Framework

Our ranking adaptation framework is built on the mixture of training data consisting of two parts: one is the source-domain training data appropriately weighted so that the cross-domain queries can be emphasized; another is the actively selected queries from target domain which tend to be those most domain-specific queries with rank labels assigned by an oracle (e.g. human annotator). Note that our method does not distinguish whether some labeled data is initially available or not in target domain. Let f denote ranking function, and based on RankSVM [20], the overall active ranking adaptation framework incorporating the mixed training data is formulated as to minimize the *L2* norm of hyperplane parameter $\vec{w}$ and the Hinge loss on pairs:

$$\begin{aligned}&\underset{\vec{w}}{\operatorname{argmin}} \left\{ \frac{1}{2}||\vec{w}||^2 \right. \\ &+ \lambda_s \sum_{q_s \in D_s} \sum_{i=1}^{\ell_{q_s}} W(q_s) * \left[1 - z_i * f\left(\vec{w}, \vec{x}_i^{(1)} - \vec{x}_i^{(2)}\right)\right]^+ \\ &\left. +\lambda_t \sum_{q_t \in D_t} \sum_{j=1}^{\ell_{q_t}} \left[1 - z_j * f\left(\vec{w}, \vec{x}_j^{(1)} - \vec{x}_j^{(2)}\right)\right]^+ \right\} \end{aligned} \quad (2)$$

where $z_i = \begin{cases} +1, & \text{if } x_i^{(1)} \succ x_i^{(2)}; \\ -1, & \text{otherwise} \end{cases}$ is the binary value depending on the ground-truth order of two documents in the i-th pair, the Hinge loss terms in $[.]^+$ correspond to the source-part loss and target-part loss based on source training queries in D_s and the actively selected target training queries in D_t ($|D_s| \gg |D_t|$), respectively, $W(q_s)$ is the importance weight of source query q_s, ℓ_{q_s} and ℓ_{q_t} are respectively the number of document pairs in source query q_s and target query q_t, and λ_s and λ_t are free coefficients balancing the volumes of the two sets. Generally, we set $\lambda_t \gg \lambda_s$ because $|D_t|$ is much less than $|D_s|$ and the training should focus on the ranking accuracy of target domain.

We first describe our query weighting scheme in Section 5.2, and then present our framework to combine adaptation with active selection in Section 5.3.

5.2 Query Weighting

Intuitively, if the retrieved documents associated with a source query q_s can be ranked well by the target ranking model which is learned on target training data only, it implies that the document information in q_s is more likely to be consistent to the target domain, thus should be deemed as more important. Otherwise q_s may be incompatible with target domain. Based on this intuition, $W(q_s)$ can be estimated with the ranking performance on q_s produced by the target ranking model. Since the source data are all labeled, we can directly use Normalized Discounted Cumulative Gain (NDCG) [21] as performance measure, which is calculated as follows:

$$W(q_s) = \frac{1}{Z_n} \sum_{i=1}^{n} \frac{2^{r(i)} - 1}{\log(1+i)} \quad (3)$$

where $r(i)$ denotes the rank label of the i-th document in the ranked list, n is length of the ranked list and Z_n is a normalization constant which is chosen so that the perfect list gets score of 1. The procedure of ranking adaptation based on query weighting is given in Algorithm 2.

Algorithm 2 Ranking adaptation based on query weighting

Input:
D_s: Training data in source domain, $|D_s| > 0$
D_t: Training data in target domain, $|D_t| > 0$
λ_s, λ_t: The free balance coefficients
Output:
Ranking model H for target domain;
1: Learn ranking model H_t on D_t;
2: **for** $i = 1$; $i \leq |D_s|$; $i++$ **do**
3: Calculate $W(q_s^i)$ by H_t using Equation 3;
4: **end for**
5: Train H on D_s and D_t with $W(.)$, λ_s and λ_t;
6: **return** H;

Note that in query weighting we assume certain labeled queries in target domain have been available for training the target ranking model. In practice, however, there can be no training data in target domain at the beginning, which is to be selected actively from scratch using the source-domain data (see the next Section).

5.3 Active Adaptation

Active adaptation algorithm should select the most informative target queries to annotate for obtaining domain-specific knowledge. Interestingly, we find the selection can be done with the help of source training queries that are weighted properly as discussed earlier. The soundness of our approach is due to the following. Note that the cross-domain queries in source domain are highlighted by weighting. It can be expected that the model built on these weighted training data can rank the cross-domain queries in target domain much better than the target domain-specific queries. In other words, the rank of target domain-specific queries (the queries we attempt to identify) tend to be less predictable, therefore leading to greater disagreement among the committee members. Therefore, the committee encourages the selection of these informative queries with domain-specific knowledge. The details of the procedure are presented in Algorithm 3.

For building the committee, which exhibits adequate diversity in terms of different generalization power of its member models, we need to resample the two sets in step 2 and 3. And then we train a ranking model on each combined subset in step 4, which makes the committee covered at utmost by the training patterns in D_s and D_t. The informative query q_i is selected in step 5. Since D_t is augmented stepwise and each committee member is built on D_s and the up-to-date D_t, the information of each new labeled q_i does not overlap with that in D_s and D_t. Finally, the final ranking model H is trained in step 9. Note that the committee is trained on the mixture of subsets of two domains, which does not change its tendency of choosing domain-specific queries from target domain.

Algorithm 3 Active query selection for ranking model adaptation

Input:

D_s: Training data in source domain, $|D_s| > 0$
D_t: Training data in target domain, $|D_t| \geq 0$
D_u: Unlabeled queries in target domain
N: The number of queries need to be labeled
T: The size of committee

Output:

Ranking model H for target domain;

1: **for** $i = 1; i \leq N; i++$ **do**
2: Sample D_s uniformly with replacement, obtain subsets $S_{s_1}, S_{s_2}, \ldots, S_{s_T}$;
3: Sample D_t uniformly with replacement, obtain subsets $S_{t_1}, S_{t_2}, \ldots, S_{t_T}$;
4: Train on each $S_{s_i} + S_{t_i}$ using Algorithm 2 and obtain committee members $h_1, h_2, \ldots, h_T$;
5: Find $q_i \in D_u$ whose ranking most disagree among $h_1, h_2, \ldots, h_T$;
6: Obtain the rank label of each document in q_i;
7: $D_t \leftarrow D_t + q_i$, $D_u \leftarrow D_u - q_i$.
8: **end for**
9: Train H on D_s and D_t using Algorithm 2;
10: **return** H;

Table 1: The statistics of the two data sets used for adaptation experiments according to different domains

Data set	Domains	# queries	# docs/query
LETOR3.0	HP03	150	1000
	NP03	150	1000
	NP04	75	1000
	TD04	75	1000
Yahoo!	Y!Large	29,921	22.7
	Y!Small	6,330	26.3

The main differences between the in-domain active rank learning (see Algorithm 1) and the active ranking adaptation (see Algorithm 3) are twofold. First, the former needs a set of initially labeled target queries to start the active learning process, and this is not necessary for the latter. Second, in contrast to the in-domain active rank learning, active ranking adaptation can make full use of the prior knowledge from source training data. At a high level, the relationship between ranking adaptation based on query weighting (see Algorithm 2) and active ranking adaptation resembles the connection between general supervised learning and active learning in that the former randomly selects instances to label and the latter only selects the informative ones.

6. EXPERIMENTS AND RESULTS

6.1 Data Sets

We used two data sets for our experiments: LETOR3.0[1] and Yahoo! Learning to Ranking Challenge[2] data sets. The statistics of two data sets are summarized in Table 1.

[1] `http://research.microsoft.com/en-us/um/beijing/projects/letor/`
[2] `http://learningtorankchallenge.yahoo.com/datasets.php`

LETOR3.0 was constructed from the raw document collections of TREC 2003 and 2004 Web track. The raw data were preprocessed into the standard format for learning to rank [30]. In LETOR3.0, each query-document pair is represented by 64 features, including both low-level features such as term frequency, inverse document frequency and document length, and high-level features such as BM25, language-modeling, PageRank and HITS. The relevance judgments take binary values, i.e., 1 (relevant) and 0 (irrelevant). In TREC's Web track, three ranking tasks were defined, namely Home Page Finding (HP), Named Page Finding (NP) and Topic Distillation (TD) [38, 39]. HP aims to return the homepage of the specific organization or person. NP is required to return the page whose name is exactly the query. TD is to return a list of entry points of good websites that contain the contents relevant to the topic. We regard each query task as a different but related domain.

Yahoo! Learning to Ranking Challenge held a track for transfer ranking. This track provided two sets of rank training data of search engines, each from a different country. The larger set (denoted as Y!Large) serves as source domain, and the smaller one (denoted as Y!Small) as target domain. There are total 700 features provided, but the feature definition is not disclosed to public. Some of the features are defined for source or target domain only while some others are defined for both. The relevance judgment of each query-document pair takes 5 levels from 0 (irrelevant) to 4 (perfectly relevant).

6.2 Setup

With LETOR data, since HP and NP are similar tasks but TD is rather different, we conducted experiments on HP03-to-NP04 and NP03-to-TD04 adaptation, where the former setting is for adapting to a similar domain and the latter for adapting to a distinct one. The aim is not only to compare the effectiveness of different algorithms, but also to examine whether the adaptation can be done between domains with various degree of divergence. With Yahoo! data, we simply follow the original setting of the Challenge, i.e. Yahoo!Large as source and Yahoo!Small as target.

We partitioned each set of the data in target domain into two parts, one part as the unlabeled pool for active query selection and the other used for evaluation only. In each round of active selection, we set the sample size as the number of currently labeled queries, i.e. $m\% = 100\%$. For efficiency, we constructed only 2 members in a committee. The free parameters λ_s and λ_t were set such that $\frac{\lambda_s}{\lambda_t}$ is inversely proportional to $\frac{|D_s|}{|D_t|}$.

Three baselines were studied comparatively. The first is the most fundamental one that *randomly* selects *target* queries for annotation, denoted as `Random-T`. This is the classical passive rank learning setting. The second is the pure active rank learning in a single domain, which *actively* selects *target* queries based on QBC using Algorithm 1, denoted as `Active-T`. Note that the first two baselines did not exploit source-domain prior knowledge. The third one is a typical ranking adaptation approach that is trained on *source* training data and the *randomly* sampled *target* training queries for *adaptation* based on the proposed query weighting scheme using Algorithm 2, denoted as `RandomAda-S-T`. Our proposed active ranking adaptation in Algorithm 3 is also trained and denoted by `ActiveAda-S-T`. Additionally, the *super-naive* adaptation method is performed for

reference where the model trained on source data is directly used for target domain.

The ranking performance was measured by NDCG@10 (see Equation 3) and Expected Reciprocal Rank (ERR) [7]. ERR is computed as follows:

$$ERR = \sum_{i=1}^{n} \frac{1}{i} R(r_i) \prod_{j=1}^{i-1} (1 - R(r_j)) \quad (4)$$

where r_i is the relevance level of document ranked at the i-th position, $R(r) = \frac{2^r - 1}{2^{r_{max}}}$ is a mapping from relevance level to probability of relevance (r_{max} is the maximum relevance level), and n is the length of document list.

6.3 Results and Discussions

In this section, we present and discuss the experimental results on the three ranking adaptation settings. The learning curves of different algorithms on HP03-to-NP04, NP03-to-TD04 and Y!Large-to-Y!Small ranking adaptation are shown in Figure 2, 3 and 4, respectively. As we can see, there are some general observations based on all three sets of results shown in the figures.

1. Including more training data from target domain is clearly more helpful to all the algorithms except for the super-naive adaptation (the straight lines in each figure) which does not use any target data. However, with more and more training data selected from target domain, different selection strategies tend to converge to some similar effectiveness. We are particularly concerned with their performance gap when not many queries are selected, which manifests the advantage of active selection that focuses on obtaining minimum number of examples.

2. `Active-T` consistently outperforms `Random-T`. T-test demonstrates that the improvement is statistically significant at most of the check points (p-value less than 0.03). This indicates that our query selection strategy based on query-by-bagging using vote entropy as ranking disagreement measure can choose informative queries much more effectively than the random selection. Note that `Active-T` must start from some initial set of labeled queries, for which we randomly choose 5 labeled queries.

3. `RandomAda-S-T` and `ActiveAda-S-T` respectively outperform `Random-T` and `Active-T`. T-test indicates that both of the improvements are significant most of the times with $p < 0.03$ and $p < 0.05$, respectively. This implies that the relevant ranking knowledge from the related domain is helpful to improve the ranking tasks in target domain, which can be successfully achieved by using our proposed source query weighting measure based on NDCG that reflects how well the source queries can be ranked by the target model.

4. `RandomAda-S-T` and `Active-T` are basically comparable. Either one is not consistently better or worse than the other. This implies that the contribution of random in-domain ranking knowledge plus out-of-domain knowledge, which are largely imprecise but abundant, tends to be equivalent to that of using a small amount of accurate in-domain knowledge.

5. Most importantly, `ActiveAda-S-T` outperforms all other algorithms and the trend is evident especially when not so many target queries are selected. T-tests shows that it significantly outperforms `Random-T` ($p < 0.01$), `Active-T` ($p < 0.03$) and `RandomAda-S-T` ($p < 0.05$) most of the times. This implies the cost-effectiveness of our active ranking adaptation approach using out-of-domain relevant knowledge to help the selection of target domain-specific queries. That is, with limited amount of budget for annotating the selected queries, our method performs the most effectively.

Our underlying assumption is that the prior training data is freely available from source domain. This follows the general presupposition of domain adaptation that ignores the original expense on collecting source data. However, from the perspective of active learning practitioners, the cost of out-of-domain training data seems not completely negligible as they might be purchased or built-in-house with initial expenses. How to take into account such cost is an open question. We believe that the initial cost is dissolved gradually as the data are reused repeatedly. Therefore, it's fair for us to ignore it here and leave the issue for future study.

6.3.1 HP03-to-NP04 Adaptation

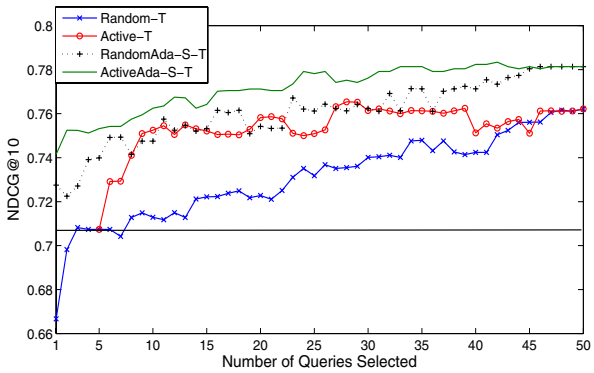
The learning curves of the algorithms under HP03-to-NP04 setting are shown in Figure 2(a) and Figure 2(b) based on NDCG@10 and ERR, respectively. Several particular observations in this setting are noteworthy.

Indeed the two domains are similar evidenced as the generally large NDCG@10 and ERR scores even with the super-naive adaptation method. This suggests that the model trained only using HP03 performs not so bad for ranking on NP04 as compared to ranking model trained directly on NP04 using limited number (say around 15) of randomly sampled queries. Furthermore, because of the close similarity between the two domains, randomly introducing new queries from target domain does not significantly disturb the original data distribution of source domain. As we can observe, therefore, the increasing trend of the performance of `RandomAda-S-T` keeps relatively smooth and is almost always superior than `Active-T` because the purposeful selection is not an advantage under this situation.

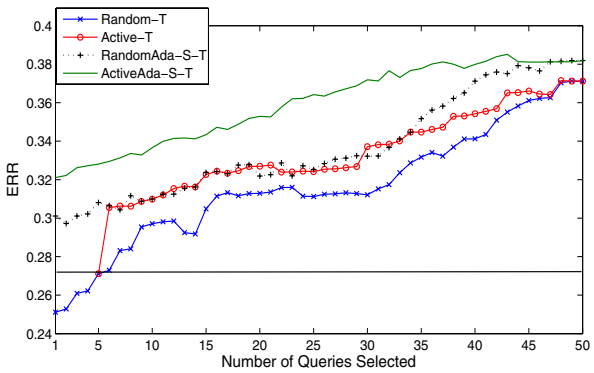
6.3.2 NP03-to-TD04 Adaptation

The learning curves of the algorithms under NP03-to-TD04 setting are shown in Figure 3(a) and Figure 3(b) based on NDCG@10 and ERR, respectively.

Compared to HP03-to-NP04 adaptation above, the two domains here have large data distribution gap. Therefore, it is more difficult for adaptation algorithms to achieve high NDCG@10 and ERR scores. Meanwhile, due to the large distribution gap, randomly introducing new queries from TD04 tends to disturb the distribution of NP03 rendering its predictive power unstable. This is clearly visible when not many random TD04 queries are used. In Figure 3(a) and Figure 3(b), the performance of `RandomAda-S-T` demonstrates some large variations when less than 15 TD04 queries are selected. Since this small number of instances are randomly chosen which may not be the domain-specific ones, they may not be able to help positively the weighting of source queries, which further harms the adaptation. As a consequence, `RandomAda-S-T` occasionally performs even worse than `Random-T`.

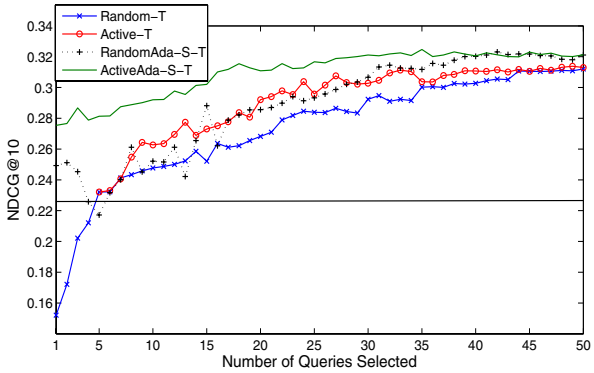


(a) NDCG@10 results

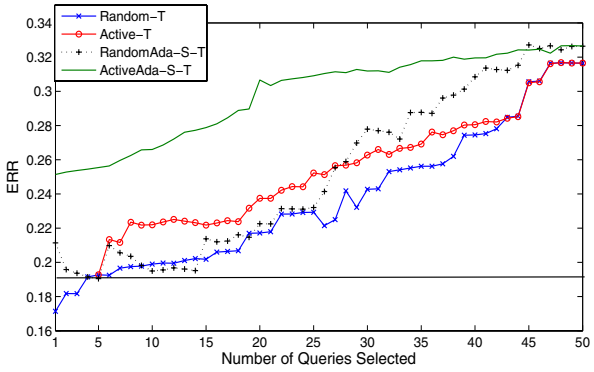


(b) ERR resutls

Figure 2: The performance comparison of different algorithms for HP03-to-NP04 ranking adaptation using LETOR3.0 dataset. The source domain is Home Page finding task in TREC-2003 (HP03) and the target domain is Named Page finding task in TREC-2004 (NP04). `Random-T`: Random query selection in target domain only; `Active-T`: Active query selection in target domain only; `RandomAda-S-T`: Random selection of target queries for ranking adaptation; `Active-S-T`: Active selection of target queries for ranking adaptation. The horizontal straight line is the performance of super-naive adaptation method which is trained only with source-domain training data.



(a) NDCG@10 results



(b) ERR results

Figure 3: The performance comparison of different algorithms for NP03-to-TD04 ranking adaptation using LETOR3.0 dataset. The source domain is Named Page Finding task in TREC-2003 (NP03) and the target domain is Topic Distillation task in TREC-2004 (TD04).

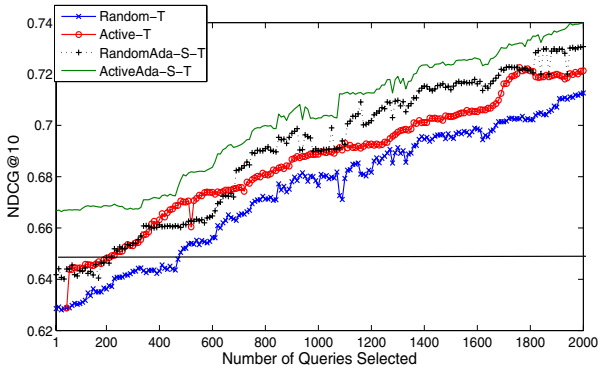
However, with more and more queries in target domain used, the target ranking model for query weighting can be radically enhanced, which then boosts the power of adaptation with the relevant prior ranking knowledge. With more than 30 target queries selected, `RandomAda-S-T` can outperform `Active-T` since `Active-T` cannot take advantage of source queries.

`ActiveAda-S-T` does not suffer from this random factor since the target queries are selected purposefully to provide the domain-specific knowledge, which aims to reinforce the weighting of source queries so as to boost up adaptation performance very quickly with minimum cost.

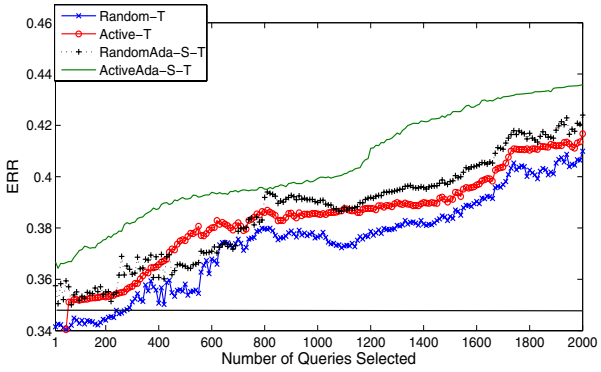
6.3.3 Y!Large-to-Y!Small Adaptation

This set of experiments examine the ranking performance in the Y!Large-to-Y!Small adaptation setting. Note that the feature space of the two domains are different. We did not particularly deal with the feature difference since our method is instance-based adaptation focusing on the selection of instances. We just naively expanded the feature space to include all features of both domains where the missing features were given as 0. We leave the feature-based study for future work. The learning curves of the algorithms are shown in Figure 3(a) and Figure 3(b) based on NDCG@10 and ERR, respectively. Here we discuss some particular observations with this setting.

On Yahoo! data set, we find that `RandomAda-S-T` outperforms `Active-T` most of the times. This data set is characterized as the large scale source training data whose size is approximately 5 times of the target data. The results indicate that the large source data is very useful for the ranking adaptation as well as query selection in the target domain.



(a) NDCG results



(b) ERR resutls

Figure 4: The performance comparison of different algorithms for Y!Large-to-Y!Small ranking adaptation using Yahoo! Learning to Rank Challenge dataset. The source domain is the large set (Y!Large) and the target domain is the small set (Y!Small).

The explanation is that with lots of labeled data, more cross-domain knowledge can be encoded in the source domain. It is more likely that these cross-domain relevant knowledge in source domain are even more helpful than those most informative target queries identified by `Active-T`.

Furthermore, unlike on LETOR where the performance of `ActiveAda-S-T` converges more quickly than others, we do not observe any trend of convergence up to the point with 2000 selected queries. This is because even the smaller target domain on Yahoo data (Y!Small) contains a lot more queries than LETOR does. Therefore, we can deduce that if with very limited budget, a small unlabeled pool of target data is a reasonable choice to do active adaptation, but if with some large budget, we should use a large unlabeled pool to take the most advantage out of active adaptation.

7. CONCLUSION AND FUTURE WORK

We propose a general rank learning framework by using the combination of active query selection and ranking model adaptation that are rooted from inherently complementary learning paradigms. The idea is that active selection focuses on providing concise knowledge while adaptation encourages the transfer of prior relevant knowledge, and the combination leverages both to improve the overall cost-effectiveness of rank learning. Specifically, we extend and incorporate the essential techniques commonly used in active sampling (e.g. query-by-committee) and domain adaptation (e.g instance weighting) into our unified iterative framework so that the two components are mutually reinforced to boost up the overall ranking performance. Experiment results show that our active ranking adaptation approach not only saves considerable amount of labeling effort but also significantly improves ranking effectiveness.

Our framework provides a general approach that can be used to combine various specific active ranking and ranking adaptation algorithms for cost-effective rank learning. Our algorithm did not outperform the champion of the Yahoo! Learning to Rank Challenge because we just applied the essential ranking techniques such as RankSVM rather than the sophisticated ones such as rank boosting trees used by others. However, the sophisticated ranking algorithms can be tried in a trivial manner. Also, we merely investigated query selection and weighting in this work, but it is expected that our method can be extended without difficulty to document level which will be studied in the future.

Acknowledgement

P. Cai and A. Zhou are supported by NSFC grant (Grant No. 60925008) and 973 program (Grant No. 2010CB731402) of China. W. Gao and K.-F. Wong are supported by national 863 program (Grant No. 2009AA01Z150) of China. We thank anonymous reviewers for their helpful comments.

8. REFERENCES

[1] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of ICML*, pages 1–9, 1998.

[2] M. Amini, N. Usunier, F. Laviolette, A. Lacasse, and P. Gallinari. A selective sampling strategy for label ranking. In *Proceedings of ECML*, pages 18–29. 2006.

[3] J. Blitzer, R. Mcdonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of EMNLP*, pages 120–128, 2006.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, August 1996.

[5] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of SIGIR*, pages 268–275, 2006.

[6] Y. S. Chan and H. T. Ng. Domain adaptation with active learning for word sense disambiguation. In *Proceedings of ACL*, pages 49–56, 2007.

[7] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of CIKM*, pages 621–630, 2009.

[8] O. Chapelle, P. K. Shivaswamy, S. Vadrevu, K. Q. Weinberger, Y. Zhang, and B. L. Tseng. Multi-task learning for boosting with application to web search ranking. In *Proceedings of SIGKDD*, pages 1189–1198, 2010.

[9] D. Chen, Y. Xiong, J. Yan, G.-R. Xue, G. Wang, and

Z. Chen. Knowledge transfer for cross domain learning to rank. *Information Retrieval*, 13(3):236–253, 2009.

[10] D. Chen, J. Yan, G. Wang, Y. Xiong, W. Fan, and Z. Chen. Transrank: A novel algorithm for transfer of rank learning. In *IEEE International Conference on Data Mining Workshops*, pages 106–115, 2008.

[11] K. Chen, R. Lu, C. Wong, G. Sun, L. Heck, and B. Tseng. Trada: Tree based ranking function adaptation. In *Proceedings of CIKM*, pages 1143–1152, 2008.

[12] I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *Proceedings of ICML*, pages 150–157, 1995.

[13] H. Daumé III. Frustratingly easy domain adaptation. In *Proceedings of ACL*, pages 256–263, 2007.

[14] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, 2006.

[15] P. Donmez and J. G. Carbonell. Optimizing estimated loss reduction for active sampling in rank learning. In *Proceedings of ICML*, pages 248–255, 2008.

[16] P. Donmez and J. G. Carbonell. Active sampling for rank learning via optimizing the area under the roc curve. In *Proceedings of ECIR*, pages 78–89, 2009.

[17] J. Gao, Q. Wu, C. Burges, K. Svore, Y. Su, N. Khan, S. Shah, and H. Zhou. Model adaptation via model interpolation and boosting for web search ranking. In *Proceedings of EMNLP*, pages 505–513, 2009.

[18] W. Gao, P. Cai, K.-F. Wong, and A. Zhou. Learning to rank only using training data from related domain. In *Proceedings of SIGIR*, pages 162–169, 2010.

[19] B. Geng, L. Yang, C. Xu, and X.-S. Hua. Ranking model adaptation for domain-specific search. In *Proceedings of CIKM*, pages 197–206, 2009.

[20] R. Herbrich, T. Graepel, and K. Obermayer. *Large Margin Rank Boundaries for Ordinal Regression*. MIT Press, Cambridge, 2000.

[21] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of SIGIR*, pages 41–48, 2000.

[22] J. Jiang and C. Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of ACL*, pages 264–271, 2007.

[23] X. Jiang, Y. Hu, and H. Li. A ranking approach to keyphrase extraction. In *Proceedings of SIGIR*, pages 756–757, 2009.

[24] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of SIGIR*, pages 3–12, 1994.

[25] H. Li, Y. Shi, M.-Y. Chen, A. G. Hauptmann, and Z. Xiong. Hybrid active learning for cross-domain video concept detection. In *Proceedings of the International Conference on Multimedia*, pages 1003–1006, 2010.

[26] N. Liu, J. Yan, D. Shen, D. Chen, Z. Chen, and Y. Li. Learning to rank audience for behavioral targeting. In *Proceedings of SIGIR*, pages 719–720, 2010.

[27] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[28] B. Long, O. Chapelle, Y. Zhang, Y. Chang, Z. Zheng, and B. L. Tseng. Active learning for ranking through expected loss optimization. In *Proceedings of SIGIR*, pages 267–274, 2010.

[29] A. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of ICML*, pages 350–358, 1998.

[30] T. Qin, T.-Y. Liu, J. Xu, and H. Li. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 2010.

[31] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, Cambridge, 2009.

[32] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, 2010.

[33] B. Settles. Active learning literature survey. Technical report, Technical Report, Computer Sciences 1648, University of Wisconsin, Madison, 2010.

[34] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 287–294, 1992.

[35] X. Shen and C. Zhai. Active feedback in ad hoc information retrieval. In *Proceedings of SIGIR*, pages 59–66, 2005.

[36] X. Shi, W. Fan, and J. Ren. Actively transfer domain knowledge. In *Proceedings of ECML and KDD*, pages 342–357, 2008.

[37] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers on large online QA collections. In *Proceedings of ACL*, pages 719–727, 2008.

[38] E. M. Voorhees. Overview of TREC 2003. In *Proceedings of TREC 2003*, pages 1–13, 2003.

[39] E. M. Voorhees. Overview of TREC 2004. In *Proceedings of TREC 2004*, pages 1–12, 2004.

[40] B. Wang, J. Tang, W. Fan, S. Chen, Z. Yang, and Y. Liu. Heterogeneous cross domain ranking in latent space. In *Proceedings of CIKM*, pages 987–996, 2009.

[41] Z. Wang, J. Feng, C. Zhang, and S. Yan. Learning to rank tags. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 42–49, 2010.

[42] Z. Xu and R. Akella. A bayesian logistic regression model for active relevance feedback. In *Proceedings of SIGIR*, pages 227–234, 2008.

[43] L. Yang, L. Wang, B. Geng, and X.-S. Hua. Query sampling for ranking learning in web serach. In *Proceedings of SIGIR*, pages 754–755, 2009.

[44] E. Yilmaz and S. Robertson. Deep versus shallow judgments in learning to rank. In *Proceedings of SIGIR*, pages 662–663, 2009.

[45] H. Yu. Svm selective sampling for ranking with application to data retrieval. In *Proceedings of SIGKDD*, pages 354–363, 2005.

[46] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of ICML*, pages 114–121, 2004.