

Active Learning to Rank Method for Documents Retrieval

Faïza Dammak, Imen Gabsi, Hager Kammoun, Abdelmajid Ben Hamadou
 MIRACL Multimedia, Information systems and Advanced Computing Laboratory,
 Technology Center of Sfax, Tunis Road Km 10, B.P. 242 Sfax 3021.
 Sfax, Tunisia

e-mail: faiza.dammak@gmail.com e-mail: imenmri@gmail.com e-mail: hager.kammoun@isd.rnu.tn e-mail: abdelmajid.benhamadou@isimsf.rnu.tn

Abstract—This paper presents a new active learning to rank algorithm based on boosting for active ranking functions. The main goal of this algorithm is to introduce unlabeled data in the learning process. Since this type of ranking is based on a phase of selection of the most informative examples to label, the proposed algorithm allows the cost of labeling to be reduced. In a first step, the algorithm proposed is going to select at each iteration the most informative query-document pair from unlabeled data using the “Query by Committee” strategy. It is this pair which maximizes the measure of disagreement between a representative committee model chosen randomly and the model generated by the supervised algorithm. In fact, the randomly chosen model is generated from the main labeled set. While the other model is generated from the labeled set which changes in each iteration, by using a supervised ranking algorithm. For the latter, we choose to use three algorithms of *boosting*: RankBoost belonging to the family of the *pairwise approach*; AdaRank and LambdaMART belonging to the family of the *listwise approach*. Our choice is meant to subsequently compare the performance of pairwise and listwise approaches. In a second step, once this pair is selected, it will be added to the labeled set. To evaluate the performance of the active model proposed, we have carried out an experimental study using the benchmark Letor 4.0 dataset. The obtained results show that the active model has a significant improvement in Normalized Discounted Cumulative Gain and Mean Average Precision.

Keywords-active learning, learning to rank, boosting ranking algorithms

I. INTRODUCTION

In front of the constant increase in the volume of information available electronically, a new field of research, dedicated to automatically optimize the ranking of results returned by systems and based on machine learning techniques, has emerged. This area of research, called learning to rank, has led to the development of many approaches and algorithms. By combining a number of existing ranking models within a single function, these approaches and algorithms have improved the quality of results lists [2].

There are three groups of learning to rank algorithms: pointwise, pairwise and listwise approaches [3]. The pointwise and pairwise approaches respectively transform ranking into (ordinal) regression or classification on single object and pairs object such as RankBoost [8]. The listwise approach [4] treats ranking lists of objects (e.g., ranking lists of documents in IR) as instances in learning, such as AdaRank [5] and LambdaMART [9], in which the group structure is considered. In our study, we focus attention on pairwise and listwise approaches: the two most successful approaches for learning to rank in IR [2].

In learning to rank, the performance of a ranking model is strongly affected by the number of labeled examples in the training set [2]. However; obtaining such information relies on human experts and hence is in general very expensive in time and in resources. Thus, we need to introduce the unlabeled data, which helps by reducing the version space size, in the training set [6].

In this article, we are interested first of all, in the problem of the reduction of the training cost of the labeled base by introducing a large unlabeled learning set as input.

We proposed an active learning to the rank algorithm which introduces a labeling process with Query-by-Committee (QBC) active learning strategy [7]. The latter has less computation than others strategies. In this method, the learner constructs a committee of classifiers based on the current training set. Each committee member then classifies the query/document pair and the learner measures the degree of disagreement among the committee members.

Nevertheless, this model used a supervised ranking algorithm to learn a ranking function. For this, we proposed three boosting algorithms using pairwise and listwise approaches: RankBoost [8] has the characteristic to consider a pair of documents as entry. While both AdaRank [5] and LambdaMART [9] use the listwise approach. This approach tries to directly optimize the value of one of the above evaluation measures, averaged over all queries in the training data. Thus, we are interested secondly in comparing the performance of pairwise and listwise approaches during the use of active learning to the rank algorithm. The applications concerned are related to the Documents

Retrieval (DR). Indeed, ranking of documents is a popular research area in IR and Web community.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 discusses the basic principles of the proposed approach. Section 4 presents the various experiments conducted to adopt the most efficient active learning to the rank model. Section 5 presents the conclusion and perspectives of this work.

II. RELATED WORK

A. Learning to Rank in DR

The main idea of learning to rank is to learn ranking functions that achieve good ranking objectives on test data. Learning to rank can be used in large variety of applications in IR. Among the typical one, we cite the DR which we take as an example in this paper. Considering a set of data compounds of query-document pairs with known relevance, the learning to rank methods learn automatically from these data the best way to combine models for optimal results list [10]. By giving a query, the ranking function attributes a score to each pair query-document. Then, this function ranks the documents in descending order of these scores. The ranking order represents the relevance of documents according to the query. This type of ranking is known as ranking of alternatives [1]. It is based on a supervised learning. However, such learning methods require a large labeled data for training. The creation of this data is generally very costly in time and resources and requires efforts from the user because it requires the intervention of a human expert. So, it is advantageous to introduce unlabeled data into the training base. The semi-supervised and active learning makes it possible to solve this problem but with different perspectives [11]. These two types of learning used a small set of labeled data and a large set of unlabeled data. By assembling both types of data, called partially labeled data, the need for labeled examples can be reduced. In the following, we present the active learning to rank approaches.

B. Active Learning to Rank

In order to get better performances and unlike the semi-supervised learning [11] which uses the unlabeled data in addition with the labeled ones, active learning puts limited human resources on labeling the most informative examples among the unlabeled ones to label [12]. This type of active learning is known as selective sampling [12] and it becomes central to many areas of applications including ranking of alternatives. On the one hand, active learning consists in learning a ranking function from a training set built during the learning and this is done by interaction with an expert. The quality of the ranking function is highly correlated with the amount of partially labeled data used to train the function. On the other hand, it proposes to the user optimal selection strategies in order to build the training set of the model [13]. The typical one is the query-by-committee (QBC) algorithm [34] which is formed by two steps. The first consists in building a committee formed by a set of *diverse* hypotheses trained on currently labeled data. The

second aims to select the optimal queries by measuring their informativeness and by calculating the disagreement among the committee members on their ranking [14] [15].

Although learning to rank has been widely studied, there are not a lot of works referring to active learning to rank [16]. Donmez and Carbonell [14] presented an active learning approach to ranking problem in the context of DR, which is in principle extensible to any other partially (or totally) ordered ranking task. The novelty of their approach lies in relying on expected loss minimization for rank learning via the use of a normalized ranking loss estimation. Long et al [17] integrate both query and document selection into active learning to rank, and propose a two-stage optimization that minimizes the expected DCG loss. Truong [18] proposed an active learning method suggested within the framework of the ranking of alternatives for the task of the text summarization. He proposed several strategies to select instances to label. Experiments have shown that they allowed to effectively forming the basis of learning by selecting the most informative instances.

III. PROPOSED APPROACH

As reported in [18], two declensions of the active ranking are cited. The first consists in selecting an entry and labeling all related alternatives. It is suitable for example for automatic summarization. The second declension seeks to select only one entry-alternative pair (query-document). The user specifies if the alternative is relevant or not in relation to this entry. This declension is particularly well adapted to applications such as the IR. In his approach, Truong [18] uses the first declension. We choose to use the second since we are interested in the field of IR, where document (alternative) and query (entry) are the components in our proposed algorithm. This algorithm uses the effective strategy to selective sampling QBC [6]. This strategy selects the element which puts in conflict most of the members of all models called committee. In our context, the most informative entry-alternative pair is the one which makes a maximum of disagreement between the committee model and the model induced on the set of alternatives by a supervised ranking algorithm. The effectiveness of this method depends on the construction of the committee which must be varied enough and representative of space of entry as well as the choice of the measure of disagreement.

A. Notation

Given a set of entry X and a set of alternatives A , we assume that each query x is associated with a subset of known alternatives $A_x \subset A$. We consider a training labeled set $S_L = \{(x_i, y_i); i \in \{1, \dots, m\}\}$ with x_i an input and y_i a set of labels associated with A_x . In addition, we consider another great set of inputs unlabeled $S_U = \{(x'_i); i \in \{m+1, \dots, m+n\}\}$. The algorithm proposed begins initially with S_L , S_U , a supervised ranking algorithm, K the number of partitions of all labeled data and Nb the desired number of examples to be labeled.

B. Active learning to rank algorithm

On the one hand, the active learning to rank algorithm (Figure 1) consists in building a committee formed by a set of diverse hypotheses trained on currently labeled data. In fact, we firstly subdivided the labeled set of training in K partitions and then associated for each partition a model. Hence; each model generates a score function h_k^{cv} as well as a score file. On the other hand, the proposed algorithm learns a model h from the labeled set. This model, changes in each iteration with the addition of a new labeled pair, by using a supervised ranking algorithm. Then, the algorithm will randomly choose a model among the K models learned at each iteration. Thereafter, it will select the most informative query-alternative pair from unlabeled data with the “Query by Committee” strategy. It is this very pair that maximizes the measure of disagreement between a representative committee model h_k^{cv} chosen randomly and the model h generated by the supervised algorithm (Figure 2). This measure is defined as follows :

$$d_c(h, h_k^{cv}, x) = \max_{l \in L_x} \{c(h, x, l) - c(h_k^{cv}, x, l)\} \quad (1)$$

where $x \in X$, L_x is the set of possible labels on alternative A_x , c is a cost function, h_k^{cv} and h two score functions. The current model asks the user to label the selected pair. Lastly, this algorithm withdraws the selected pair from S_U and adds it in S_L until reaching the desired number of labeled data. As an output, it provides the required score function.

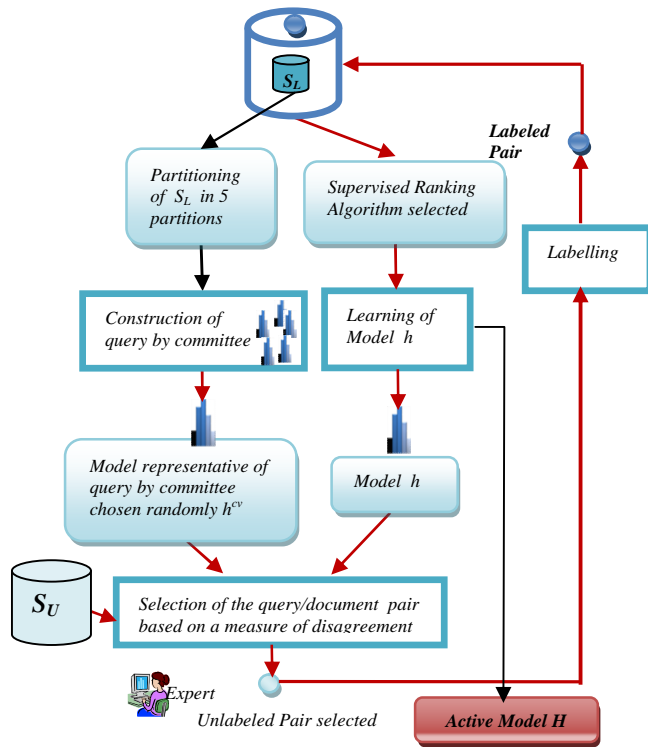


Figure 1. Approach proposed

For the supervised ranking algorithm, we have chosen three boosting algorithms: RankBoost [8], LambdaMART [9] and AdaRank [5]. RankBoost [8] is a powerful pairwise supervised learning algorithm that learns a real-valued (scoring) function, by optimizing a specific error measure suitable for ordering sets of objects. More precisely, at each round of boosting, the algorithm minimizes the weighted number of instances that are disordered. The pairs on which we have made mistakes (with respect to the weaker ranker chosen for that round) are given a higher importance weight for correct ordering in the next round. Thus, the goal of RankBoost is to produce an order, by a scoring function h , for each document, which places as many relevant documents as possible at the top. LambdaMART [19] is a listwise method; it is the boosted tree version of LambdaRank [20]. It uses Gradient boosting [21] to optimize a ranking cost. It employs the MART (Multiple Additive Regression Trees) algorithm to learn a boosted regression tree as a ranking model. LambdaMART has been shown to be among the best performing learning methods based on evaluations on public data sets [22]. Readers can refer to [26] for details of this algorithm. AdaRank [5] is a listwise algorithm for learning ranking models in DR. It repeatedly constructs ‘weak rankers’ on the basis of re-weighted training data. Finally, it linearly combines the weak rankers for making ranking predictions. In contrast to the existing methods, AdaRank optimizes a loss function that is directly defined on the performance measures. It employs a boosting technique in ranking model learning.

In the following, we give the active ranking algorithm.

<p>Algorithm. Active learning to rank algorithm of alternatives</p> <p>Entry :</p> <ul style="list-style-type: none"> - A small labeled data $S_L = \{(x_i, y_i); i \in \{1, \dots, m\}\}$ - A large unlabeled data $S_U = \{(x_i); i \in \{m+1, \dots, m+n\}\}$ - A supervised ranking algorithm - K : number of partitions of S_L - Nb : number of labeled examples required - Learn models of the committee and obtain h_k^{cv} - $nbIter \leftarrow 0$ <p>While $nbIter \leq Nb$ do</p> <ul style="list-style-type: none"> - learn a ranking function h with supervised algorithm on S_L - choose randomly h^{cv} - select the most informative query-document pair from S_U which maximizes the measure of disagreement $d_c(h, h^{cv}, (x, d))$ - ask the expert to label this pair - withdraw this pair of S_U and add it in S_L - $nbIter \leftarrow nbIter + 1$ <p>End</p> <p>Output : Active Model H</p>
--

Figure 2. Active learning to rank algorithm of alternatives

The total cost is dominated by the step of selecting the input. In our case, it is the calculation of the disagreement measure, which requires taking into account all the possible values of labels for a given input. As well, QBC strategy is easy to implement. It has a low complexity. The algorithm requires training $K+1$ models. The cost of learning is therefore multiplied by $K+1$.

IV. EXPERIMENTAL STUDY

We conducted a number of experiments in order to evaluate the importance of unlabeled data to learn an efficient ranking function. Once the ranking function is learned in the training phase, it will be used to order unlabeled examples from the test data. This training phase can be followed by a validation phase.

A. Experimental tools

Evaluating the quality of ranking functions is a core task in DR and other IR domains. For the realization of the algorithm (Figure 2), we propose to extend the library of learning to rank algorithms RankLib [23]. Currently, this library contains the implementation of eight ranking algorithms. It also provides the implementation of the evaluation measures based on the performance measures used in IR tools.

For the realization of the stage of selection of the query-document pair, we initially propose to evaluate the disagreement between a representative of committee model h_k^{cv} and the model h in order to select the unlabeled pairs. Then, we choose to use as measures of disagreement the Euclidean distance between the score given by model h_k^{cv} and the score obtained by the model h deduced from the supervised algorithm. It is the selected pair which has the maximum distance. By varying the number desired of data to label and by calculating the variation of the evaluation measures, we can deduce the suitable supervised boosting algorithm from the three chosen: RankBoost, LambdaMART, and AdaRank.

1) Data collections

Since the performance of a model depends on the quality of data used in the learning phase, we use the standard benchmark LETOR (*LEARNING TO RANK*) [1] which constitutes a baseline in IR and evaluation measures [24]. We use specially the MQ2008-semi (*Million Query track*) collections in LETOR 4.0 as it contains both labeled and unlabeled data. There are about 2000 queries in this dataset. On average, each query is associated with about 40 labeled documents and about 1000 unlabeled documents.

MQ2008-semi [25] is conducted on the .GOV2 corpus using the TREC 2008, which is crawled from Web sites in the .gov domain. There are 25 million documents contained in the .GOV2 corpus, including HTML documents, as well as the extracted text of PDF and Word and postscript files [25].

Each subset of the collection MQ2008-semi is partitioned into five divisions, denoted as S1, S2, S3, S4, and S5, in order to conduct a five-fold cross validation. The results reported in this section are the average results over multiple folds. For each fold, three parts are used: The training part is used to learn the ranking model. The validation part is used to tune the parameters of the ranking model, like the number of iterations in RankBoost. The test part is used to report the ranking performance of the model.

Also, in this semi-MQ2008 collection, each training file contains a small number of pairs of labeled data and a large number of pairs of unlabeled data. We choose to extract pairs of labeled data in a first file for the training phase and the unlabeled pairs in a second file for the testing phase. In addition, the unlabeled pairs of data will be selected and labeled by the proposed algorithm in learning and added to the labeled file extracted.

2) Evaluation Measures

For the evaluation of the algorithms proposed (RankBoost_Active, LambdaMART_Active, AdaRank_NDCG_Active and AdaRank_MAP_Active), we use a set of standard ranking measures such as Precision at position n , Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) [24]. $P@n$ measures the accuracy within the top n results of the returned ranked list for a query:

$$P@n = \frac{\# \text{relevant docs in top } n \text{ results}}{n}. \quad (2)$$

MAP takes the mean of the average precision values over all relevant documents:

$$MAP = \frac{\sum_{n=1}^N (P@n * rel(n))}{\# \text{total relevant docs for this query}}. \quad (3)$$

$NDCG@k$ is widely used to handle multiple levels of relevance (whereas Precision and MAP are designed for binary relevance levels). The value of the NDCG to a position k of ordered list is calculated as follows:

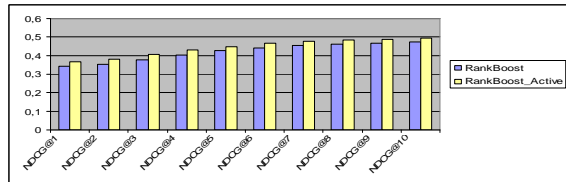
$$NDCG @ k = \frac{1}{n} \sum_{k=1}^n \frac{1}{z_{K_n}} \sum_{j=1}^k m_j \frac{2^{r(j)} - 1}{\log(1 + j^k)} \quad (4)$$

3) Experimental results

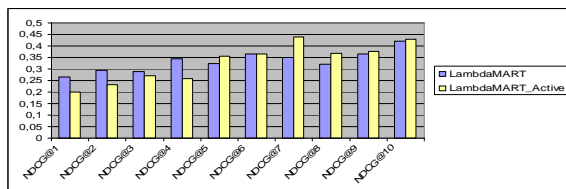
These experimental results test how unlabeled data affect the ranking performance of the proposed algorithms.

RankBoost, LambdaMART, AdaRank_MAP and AdaRank_NDCG were selected as baselines in the experiments. For the proposed algorithms, the number of iterations was determined automatically during each experiment. Specifically, when there is no improvement in ranking accuracy in terms of the performance measure, the iteration stops. For both RankBoost_Active and LambdaMART_Active, we train the ranker for 500 rounds. For the others (AdaRank_NDCG_Active and AdaRank_MAP_Active), the number of iterations was stopped at 200 rounds.

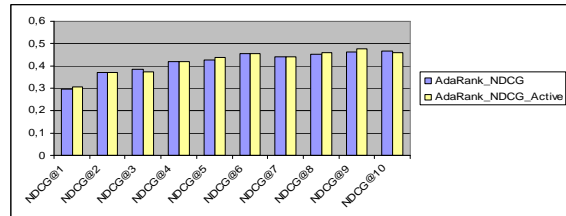
Then, we calculated the variation of NDCG for the three algorithms according to the number desired of data to label (Figure 3: (a), (b), (c) and (d)). Each group of bars corresponds to one NDCG@n. As shown in this figure and Table I, NDCG@n measures are better in RankBoost_Active algorithm, quite better in AdaRank_NDCG_Active and AdaRank_MAP_Active. But, they are variable in LambdaMART_Active algorithm.



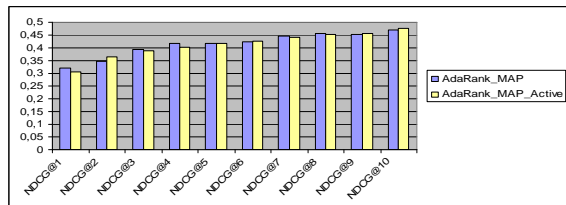
(a)



(b)



(c)



(d)

Figure 3. Performance of RankBoost_Active (a), LambdaMART_Active (b), AdaRank_NDCG_Active (c) and AdaRank_MAP_Active (d) on training set : NDCG@n measures on the MQ2008-semi collection

TABLE I. EVALUATION RESULTS IN TERMS OF NDCG@N ON MQ2008-SEMI DATA SET ON TRAINING SET

	NDCG@1	NDCG@2	NDCG@3	NDCG@5	NDCG@7	NDCG@8	NDCG@9	NDCG@10
RankBoost	0,3419	0,3543	0,3772	0,4267	0,4543	0,4640	0,4675	0,4731
RankBoost_Active	0,3672	0,3817	0,4075	0,4474	0,4764	0,4852	0,4874	0,4930
AdaRank_NDCG	0,2970	0,3704	0,3850	0,4256	0,4423	0,4522	0,4637	0,4655
AdaRank_NDCG_Active	0,3056	0,3704	0,3745	0,4365	0,4422	0,4601	0,4777	0,4581
AdaRank_MAP	0,3205	0,3458	0,3929	0,4174	0,4480	0,4561	0,4528	0,4702
AdaRank_MAP_Active	0,3065	0,3660	0,3886	0,4183	0,4416	0,4516	0,4566	0,4755
LambdaMART	0,2660	0,2951	0,2875	0,3221	0,3498	0,3194	0,3651	0,4208
LambdaMART_Active	0,2009	0,2314	0,2710	0,3552	0,4374	0,3688	0,3776	0,4293

We noticed also that, in the training and testing set, the pairwise RankBoost_Active has NDCG values slightly higher than AdaRank_NDCG_Active, AdaRank_MAP_Active and LambdaMART_Active algorithms, which belong to listwise approach, (Figure 5). Although, compared with the other two types of approaches (pointwise and pairwise), the listwise approaches express the real sense of the learning to rank. The experimental studies have shown that the pairwise approach is better for the active algorithm proposed. Indeed, pairwise ranking methods have shown their performances by balancing the distribution of document pairs across queries [2]. These results illustrate how the unlabeled data affect the performance of ranking in the proposed algorithm. We notice a slight improvement by using the criterion NDCG@n for the fourth active algorithms.

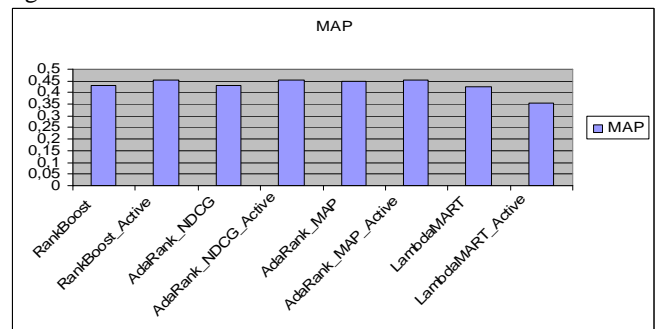


Figure 4. Performance on test set : MAP measures on the MQ2008-semi collection

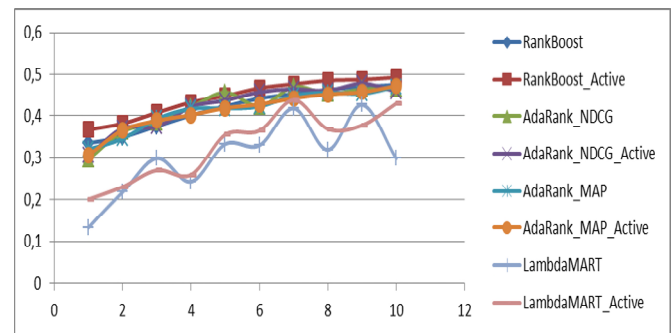


Figure 5. Performance on testing set : NDCG@n measures on the MQ2008-semi collection

Figure 4 demonstrates the results of MAP measures on the MQ2008-semi collection. The results of this figure show that RankBoost_Active, AdaRank_NDCG_Active and AdaRank_MAP_Active algorithms have an average precision (MAP) better than that found by RankBoost and AdaRank_NDCG and AdaRank_MAP (Figure 4).

These results prove the interest of integrating unlabeled data in ranking functions with active learning.

V. CONCLUSION

In this article, we have proposed an active learning to rank algorithm based on a supervised ranking one. The contribution of this algorithm is presented in the use of a very small number of labeled examples and a large number of unlabeled data preselected incrementally by the Query By Committee method. This method has been shown to be effective in different classification tasks. For supervised ranking algorithm, we have chosen three boosting algorithm for different approaches the pairwise and listwise. The training and the test phases were carried out with the collections of the benchmark standard LETOR 4.0. Basing on the measures of evaluation NDCG and MAP, the preliminary results show that the active algorithm using pairwise approach provides better results.

The performance of such a model lies in its ability to use, for training, unlabeled data and QBC method. The latter allows minimizing the version space. However, its performance degrades when the number of labeled data and the learning time increase. To solve this problem, we suggest integrating a semi-supervised learning method to label the selected pair instead of the expert to reduce the learning time.

REFERENCES

- [1] T.-Y. Liu, J. Xu, T. Qin, W.-Y. Xiong, and H. Li, "LETOR: Benchmark dataset for research on learning to rank for Information Retrieval". Proceedings of the Learning to Rank workshop in the 30th annual international ACM SIGIR conference on Research and development information retrieval, 2007
- [2] T. -Y. Liu. "Learning to rank for information retrieval". Springer-Verlag Berlin Heidelberg, 2011.
- [3] Z. Cao, T. Qin, T.-Y. Liu, Tsai, M.-F., and Li, H. "Learning to rank: from pairwise approach to listwise approach". ICML '07, 2007, pp. 129-136.
- [4] F. Xia, T. Liu, J. Wang, W. Zhang, and H. Li, "Listwise approach to learning to rank: theory and algorithm". In ICML '08, New York, NY, USA, ACM 2008, pp. 1192-1199.
- [5] J. Xu and H. Li, "AdaRank: a boosting algorithm for information retrieval". In Proceedings of the 30th Annual International ACM SIGIR Conference (SIGIR'07), Amsterdam, 2007, pp. 391-398.
- [6] B. Settles, "Active learning literature survey", Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009.
- [7] S. Jerzy and P. Mateusz, "Comparing Performance of Committee Based Approaches to Active Learning". Recent Advances in Intelligent Information Systems, 2009, pp. 457-470.
- [8] Y. Freund, R. Iyer, R. E Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences". Journal of Machine Learning Research, 2003, pp. 933-969.
- [9] C. Burges. "From RankNet to LambdaRank to LambdaMART: An overview. Technical report", Microsoft Research Technical Report MSR-TR-2010-82, 2010.
- [10] O. Chapelle and Y. Chang. "Yahoo! Learning to Rank Challenge Overview". Journal of Machine Learning Research - Proceedings Track, vol. 14, 2011, pp. 1-24.
- [11] K. Duh and K. Kirchhoff, "Learning to rank with partially-labeled data". In Myaeng, S.-H. Oard, D. W. Sebastiani, F. Chua, T.-S., and Leong, M.-K., editors, SIGIR, ACM. 2008, pp. 251-258.
- [12] Y. Freund, H. S. Seung, E. Shamir, and N. i Tishby, "Selective sampling using the query by committee algorithm," Machine Learning, vol. 28, 1997, pp. 133-168.
- [13] N. Ailon, "An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity". Journal of Machine Learning Research, 2012, pp.137-164.
- [14] P. Donmez and J. G. Carbonell, "Active samplings for rank learning via optimizing the area under the roc curve". ECIR, volume 5478 of Lecture Notes in Computer Science, Springer. 2009, pp. 78-89.
- [15] W. Shen and H. Lin, "Active Sampling of Pairs and Points for Large-scale Linear Bipartite Ranking". In Proceedings of ACML'13 (JMLR W&CP 29), 2013, pp. 388-403.
- [16] B. Qian, H. Li, J. Wang, X. Wang, and I. Davidson. "Active Learning to Rank using Pairwise Supervision". Proceedings of the 13th SIAM International Conference on Data Mining, 2013, pp. 297-305.
- [17] B. Long, O. Chapelle, Y. Zhang, Y. Chang, Z. Zheng, and B. Tseng. "Active learning for ranking through expected loss optimization". In Proceedings of the 33rd international ACM SIGIR'10 New York, USA, 2010, pp. 267-274.
- [18] T.-V. Truong, "Learning Functions ranking with little Labeled Examples", PhD thesis, University Pierre and Marie Curie – Paris VI, 2009.
- [19] Q. Wu, C.J.C. Burges, K. Svore, and J. Gao, "Adapting Boosting for Information Retrieval Measures". Journal of Information Retrieval, 2007.
- [20] J. H. Friedman. "Greedy function approximation: A gradient boosting machine". Annals of Statistics, 29: 2000, pp. 1189-1232.
- [21] Y. Ganjisaffar, R. Caruana, and C.V. Lopes, "Bagging Gradient-Boosted Trees for High Precision, Low Variance Ranking Models", SIGIR'11, Beijing, China. 2011.
- [22] C. Sawade, S. Bickel, T. Oertzen, T. Scheer, and N. Landwehr. "Active Evaluation of Ranking Functions based on Graded Relevance". ECML PKDD'12 - Volume II. Springer-Verlag Berlin, 2012, pp. 676-691.
- [23] <http://people.cs.umass.edu/~vdang/ranklib.html>
- [24] K. Jarvelin and J. Kekalainen. "IR evaluation methods for retrieving highly relevant documents". Special Interest Group on Information Retrieval (SIGIR), 2000.
- [25] [http://research.microsoft.com/en-us/um/beijing/projects/letor//](http://research.microsoft.com/en-us/um/beijing/projects/letor/)
- [26] A. J. Aslam, E. Kanoulas, V. Pavlu, S. Savev, and E. Yilmaz. "Document selection methodologies for efficient and effective learning-to-rank". In Proceedings of the 32nd international ACM, SIGIR '09, New York, USA. 2009, pp. 468-475.