

Received November 17, 2020, accepted November 29, 2020, date of publication December 15, 2020, date of current version December 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043459

Improving Query Quality for Transductive Learning in Learning to Rank

XIN ZHANG¹ AND ZHI CHENG

Department of Big Data and Information Engineering, Hefei University, Hefei 230601, China

Corresponding author: Xin Zhang (zhangxin@hfu.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61806068, in part by the Anhui Provincial Natural Science Foundation under Grant 2008085QF290, in part by the Natural Science Foundation of Hefei University under Grant 18ZR07ZDA, Grant 19ZR04ZDA, and Grant 19ZR06ZDA; in part by the Natural Science Research Key Project of Anhui University under Grant KJ2018A0556, in part by the grant of Major Science and Technology Project of Anhui Province under Grant 17030901026, and in part by the grant from the Natural Science Foundation of Hefei University under Grant 18-19RC27.

ABSTRACT In traditional transductive learning, all queries are used in learning to rank in order to generate pseudo-labels when sufficient training data are not available. However, low quality queries may affect retrieval performance in transductive learning. We thus think that it is important to improve the quality of queries in transductive learning to train an effective ranking model. By using a small number of reliable samples and data close to the boundaries of classification, we propose building a query quality estimator by establishing a relationship between the benefits of good retrieval performance and features of the normalized query commitment that influence query quality. In our proposed transduction model, all queries available are filtered by the proposed query quality estimator and only high quality queries that enhance the effectiveness of retrieval such that they yield performance-related benefits, are used to generate pseudo-labels for learning to rank. Queries that can degrade performance benefits are discarded while creating the pseudo-labels. Pseudo-labels aggregated by high quality queries in transductive learning are then leveraged in learning to rank scenarios without sufficient training data. The results of extensive experiments on the standard LETOR 4.0 dataset showed that our proposed method can outperform strong baselines and the average normalized discounted cumulative gain is enhanced up to 7.77% in some case.

INDEX TERMS Transductive learning, query quality, retrieval performance, learning to rank.

I. INTRODUCTION

Several techniques have been proposed in recent decades to construct ranking models for information retrieval, including traditional heuristic methods, probabilistic methods, and machine learning methods [1], [2]. Of these, learning to rank is among the most popular weighting schemes [3]. However, the need for a sufficient amount of training data in learning to rank renders its use expensive and unfeasible when it is difficult to gather labels for the data [4].

Transductive learning [5]–[7], a semi-supervised mode of learning, is often used to iteratively aggregate pseudo-labels for learning to rank in information retrieval in case a sufficient amount of training data are not available [8]. In information retrieval, it is often assumed that the top-ranking documents in the initial retrieval results, namely the results of

content-based retrieval, are highly relevant to the given query while the bottom-ranking documents are irrelevant [9], [10]. Thus, the top-ranking documents of each query are taken as positive samples and the bottom-ranking documents as negative ones in transduction. However, the assumption of transductive learning is not always reliable. For example, only 13% of the five top-ranking documents in the initial retrieval results on MQ2008, a subject of LETOR 4.0 [11], were found to be relevant to the queries. In this case, noise was introduced to the pseudo-positive examples when transductive learning was applied, which significantly degraded retrieval performance. Therefore, to guarantee the effectiveness of learning to rank in information retrieval, it is necessary to enhance the quality of the queries and their associated pseudo-labels during transductive learning.

Considering that a small amount of labeled samples can be easily obtained in most application scenarios, we aimed to improve query quality by selecting pseudo-data on a

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhou¹.

per-query basis. We considered these labeled samples reliable as they are manually annotated rather than generated by a semi-supervised algorithm. Moreover, examples with a low degree of confidence (unconfident examples), located close to the boundaries of classification, significantly affect the effectiveness of retrieval [12] and these unconfident examples are incorporated in our methods. To improve the quality of pseudo-labels in transduction, we estimate the quality of each given query by determining whether it can enhance retrieval performance with a limited number of reliable labels and data with low confidence close to the classification boundaries. High quality queries are extracted by building a query quality estimator between quality-related features based on the normalized query commitment (NQC) and performance-related benefits. The queries used for training are classified into two groups, a high quality and a low quality group. Only queries that can improve performance were incorporated into the transduction to create pseudo-labels for learning to rank while those belonging to the low quality group are discarded.

The major contributions of this paper are two-fold. First, we proposed improving the effectiveness of retrieval of transductive learning by building a query quality estimator that uses a small number of reliable examples as well as examples with low confidence located close to the classification boundaries. In contrast to traditional transductive learning, only high quality queries were leveraged to aggregate pseudo-examples during the iterative process of transduction. Second, experiments on a standard dataset showed that the automatically learned features for ranking may not be sufficient to improve the effectiveness of retrieval in complex ranking models while the improvement in the quality of the training data played an important role in enhancing the effectiveness of information retrieval.

The remainder of this paper is organized as follows: Section II explains the application of transductive learning in information retrieval and surveys research on improving training data for learning to rank. Section III presents the proposed transductive learning approach in detail, and Section IV introduces the experimental settings and explains the baselines employed in the evaluation for comparison with the proposed method. Section V shows the experimental results on LETOR 4.0 and Section VI gives discussions. Section VII offers the conclusions of this work.

II. RELATED WORK

A. TRANSDUCTIVE LEARNING IN INFORMATION RETRIEVAL

Semi-supervised learning has been applied to many domains, such as indoor localization, visual ranking, speech recognition, scintillation detection, and web classification [7], [13]–[17]. For example, to enhance recommendation performance, Zhang *et al.* proposed a graph-based semi-supervised learning algorithm for indoor localization by utilizing crowd-sourced data [13]. To reduce the word error rate in speech recognition, semi-supervised learning was used

to improve model generalization with limited available data [14]. For visual ranking, a semi-supervised learning algorithm, the SSLPP, was developed by incorporating information into the degree of relevance [15]. For scintillation detection, a semi-supervised detection system based on the DeepInfomax approach was developed [16]. Besides, Semi-supervised learning can be combined with active learning to improve the efficiency of classification. Stikic *et al.* proposed combining self-training, co-training, and active learning for human activity recognition [17], and Wang *et al.* proposed a semi-supervised learning algorithm that combines active learning with transductive SVM for applications without sufficient labeled data [7]. For learning to rank without sufficient labeled data in information retrieval, many studies have verified the benefits of using transductive learning [4], [8], [17], [18].

Transductive learning, a self-training-based semi-supervised learning approach, has been widely used in applications where there only a small amount of labeled data are available, or none are available at all [5]. In the scenarios mentioned above, transductive learning is often utilized to create pseudo-labels for learning to rank in information retrieval, where it needs to only train a learning model to predict the pseudo-label of the given set in test examples, instead of all unobserved examples, as explained in [6]. The algorithm iteratively generates pseudo-examples from the remaining unlabeled data to construct a pseudo-training set for learning to rank algorithms. This section briefly explains the general setup of transductive learning in information retrieval, as described in [7], [8], and [18].

- Extraction of initial pseudo-labels. The top-ranking and bottom-ranking documents in the initial results of content-based retrieval are chosen as the initial positive and negative labels, respectively. The remaining documents in the initial results of retrieval are taken as unlabeled examples.
- Construction of a learning model. A classifier is built by making using of the initial pseudo-labels.
- Selection of new pseudo-labels. The classifier learned above is used to classify the remaining unlabeled examples, and more pseudo-relevant and irrelevant examples are added to the original training data.
- Reconstruction of a learned model. A new learned model is trained by utilizing the most recently updated pseudo-labels.

The processes of the selection of the new pseudo-labels and reconstruction of the learning model are iterative, and continue until the halting criterion has been satisfied. For learning to rank applications where human labels are rare or unavailable, the pseudo-labels are generated iteratively by the transduction-based approach described above [19], [20].

The effectiveness of the transductive learning approach depends mainly on the quality of the pseudo-examples, which in turn is affected directly by query quality. However, the traditional transductive method does not consider the quality of individual queries, and its retrieval performance is affected if

queries with poor quality are included in the training query set for learning to rank.

B. IMPROVING QUALITY OF TRAINING DATA

The quality of the training data has a significant impact on retrieval performance in information retrieval [21]–[24]. Previous research on the quality of the training data has focused mainly on supervised learning, including supervised classification and supervised learning to rank [25]–[27]. Research on handling noise-related data in classification can be divided into three categories: methods of feature selection, data selection, and establishing noise-tolerant models. In this paper, we present research on improving the training data in learning to rank.

Given that the click logs of a web user can accurately reflect the relevance of a given document to the corresponding query, Xu *et al.* used document click logs to correct noisy data for learning to rank [24]. Insisting that not all labels of the training data are reliable, they proposed two dependent models (a sequentially dependent model and a fully dependent model) to predict the pseudo-labels of the training data again. If the predicted label and the original label of a document were significantly inconsistent, the data was judged to be noisy and their labels were corrected manually. Carvalho and Elsas proposed the sigmoid loss function to replace the hinge function in RankSVM to reduce the impact of noise on document pairs [25]. Geng *et al.* proposed a feature selection method in which the relevant documents to a given query are re-ranked to ensure the hierarchy of the query and the documents in learning to rank [26]. Geng *et al.* proposed a probabilistic graphical model by introducing an implicit variable to identify the true annotation of a document [27].

In the context of semi-supervised learning, some research has been published on improving the quality of the training data. Mallapragada *et al.* proposed an active query selection method by using the min–max criterion for semi-supervised clustering [28]. Considering that graph-based methods are limited in their ability to jointly model graph structures and data features, Wu *et al.* proposed a graph-filtering framework that injects graph similarity into the data features by considering them to be signals on the graph and applying a low-pass graph filter to extract useful data representations for classification, where labels can be efficiently assigned by conveniently adjusting the strength of the graph filter [29]. Some research has also considered filtering the training queries for semi-supervised learning in learning to rank [19]–[21]. Rahangdale and Raut proposed a clustering-based semi-supervised learning method and combined it with the non-measure-specific listwise approach for learning to rank in case no labeled data are available [19]. Zhang *et al.* proposed a two-step clustering approach to filter the quality of the query in semi-supervised learning for learning to rank in case no labeled data are available [20]. Muandet *et al.* selected the example which led to the largest perturbation in the labels of the other examples, and used active learning to query a label for an unlabeled data [21].

We think that the quality of pseudo-labels should be enhanced in transductive learning. Besides, considering that a small number of reliable samples and unconfident examples are obtained in our application, we propose to improve the quality of the pseudo-labels by constructing a query quality estimator using semi-supervised learning algorithms instead of unsupervised learning.

III. FRAMEWORK OF PROPOSED TRANSDUCTIVE LEARNING WITH LIMITED RELIABLE LABELS AND EXAMPLES WITH LOW CONFIDENCE

In this section, we provide details of the proposed transductive learning method that learns a query quality estimator based on a few reliable examples, and examples with low confidence located close to the classification boundaries (unconfident examples) to select high-quality queries. All queries available are filtered by the query quality estimator, thus high quality queries are selected from these candidate queries. These selected queries were utilized to generate pseudo-labels iteratively for learning to rank. In general, our proposed algorithm was composed of three phases:

- estimating query quality with a few reliable examples;
- tagging the unlabeled data with transductive learning, where only high quality queries were utilized;
- retrieval documents by learning to rank and generating the results of retrieval using the tagged pseudo-labels for learning to rank.

The last phase is a common step in semi-supervised learning to rank, in which the pseudo-labels generated by transduction are used as input in learning to rank to train a ranking model for predicting the relevance of each document of a given query. The framework of our proposed approach is illustrated in Fig. 1.

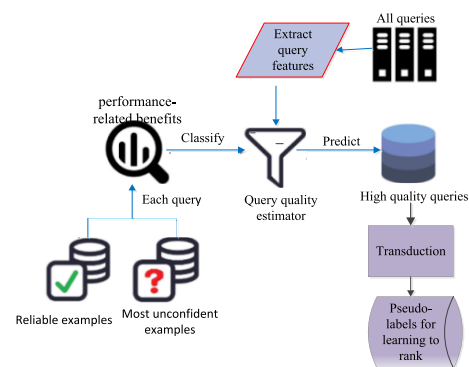


FIGURE 1. The framework of the proposed transductive learning approach.

A. ESTIMATING QUERY QUALITY

The quality of queries in a dataset differs. The relevant documents obtained for some queries were highly ranks in the initial (content-based) retrieval results, whereas other queries may have few relevant documents scattered around the initial results. Queries with high-ranking relevant documents were

believed to have a high mean average precision, and were defined as high quality in this paper. In contrast, queries with few relevant documents were considered low quality as they had a low mean average precision. Our proposed method aims to improve transductive learning involved selecting high quality queries.

1) FEATURES OF QUERY QUALITY

To estimate query quality, we needed to extract features that could indicate it. Query quality prediction, the task of estimating the effectiveness of a retrieval system given a search query in the absence of any feedback from the searcher [30], has been proven to be very challenging [30]–[33]. The methods of estimating query quality are diverse, such as clarity [31], weighted information gain (WIG) [32], NQC [33], and QPP for microblog search [30]. As in [21], [22], we used NQC in our method because it was easy to compute while still being effective in estimating query performance, as reported in [33]. The NQC measures the amount of query drift in the results list. As shown in [33], NQC is computed as follows:

$$NQC(Q) = \frac{\sigma_R}{|Score(Q, C)|} \quad (1)$$

where R is the commitment of documents related to query Q , σ_R is the standard deviation of the retrieval scores of the documents in R , and $Score(Q, C)$ is the retrieval score of the collection.

Considering that the datasets we used had different fields, including title, body, anchor, URL, and the entire given document, we extracted query features for different fields: query features based on title, body, anchor, URL, and the whole document. We also collected the content-based scores of the corresponding fields by applying TF*IDF, BM25, LMIR.ABS, LMIR.DIR, and LMIR.JM, and then computed the NQC scores of the different query features. Therefore, in our proposed method, 25 query features were used to indicate query quality, as listed in Table 1.

2) SELECTING HIGH QUALITY QUERIES

Queries used to represent different users' information-related needs had unique characteristics [34]. Some queries were popular and the top-ranking retrieval results could supply informative messages, whereas some queries were not popular and the retrieval results of these queries were not satisfactory. Intuitively, popular queries may be highly precise in ranking documents, and thus have a high mean precision. We called these queries high quality queries; they were expected to provide pseudo-examples with little noisy data.

During the selection of high quality queries, all queries available were divided into three categories without overlap: training, validation, and testing queries. For each query-document pair, we exploited query features on the basis of the NQC to represent its quality. Note that it was assumed that a small number of reliable examples could be obtained for each query-document pair to train a query dataset. We first trained a *ranker* by utilizing a few reliable samples, and then

TABLE 1. NQC features used to estimate query quality.

Fields	Retrievals model
Query quality features of title	TF*IDF
	BM25
	LAIR.ABS
	LMIR.JM
	LMIR.DIR
Query quality features of body	TF*IDF
	BM25
	LAIR.ABS
	LMIR.JM
	LMIR.DIR
Query quality features of anchor	TF*IDF
	BM25
	LAIR.ABS
	LMIR.JM
	LMIR.DIR
Query quality features of URL	TF*IDF
	BM25
	LAIR.ABS
	LMIR.JM
	LMIR.DIR
Query quality features of whole document	TF*IDF
	BM25
	LAIR.ABS
	LMIR.JM
	LMIR.DIR

greedily added a small number of samples with low confidence randomly to the original training data to re-construct the learning ranker. Because the LETOR 4.0 dataset, with about 1,700 queries on MQ2007 and 800 queries on MQ2008, was large, the enumeration-based sampling of examples with low confidence located close to the classification boundaries was infeasible. Thus, random sampling was conducted 10 ~ 15 times, and achieved near optimal results. Finally, the learning ranker with the highest mean average precision was chosen, and its effectiveness was tested on the validation queries. We then obtained the mean average precision α by applying the ranker to the validation queries. Before the ranker was applied to the validation dataset, a content-based retrieval model was used on the validation queries to obtain a mean average precision β . We thus obtained a subtracted value of mean average precision $\delta, \alpha - \beta$. A query quality estimator was then built by using the query-related features and the difference in mean average precision (δ) between the ranker and the content-based retrieval model on the set of the validation queries set.

The query quality estimator was then applied to each test query in the test queries' collection to assess its ability to enhance retrieval performance. If a query enhanced the effectiveness of retrieval on the test queries, namely, if the mean average precision obtained by the query quality estimator was greater than zero, it was considered a high-quality query. Queries that enhanced the mean average precision were added to the high-quality queries' set. It was expected to enhance the effectiveness of retrieval when extracting the top-ranking and bottom-ranking documents. Because documents ranked high by a high quality query were likely to be mostly

TABLE 2. Query types and their roles during the selection of high-quality queries.

Query type	Composition	Role
Training	1/3 all queries	<ol style="list-style-type: none"> 1. Get a performance benefit for each query by applying the learning to rank algorithm; 2. Build a query quality estimator by utilizing the NQC features and the performance benefit.
Validation	1/3 all queries	<ol style="list-style-type: none"> 1. Test the effectiveness of retrieval of each ranker built by per training query 2. Get the best ranker with the highest mean average precision
Testing	1/3 all queries	<ol style="list-style-type: none"> 1. Get the high-quality queries. The query quality estimator is applied on the testing query set to select high-quality queries, which are leveraged by transductive learning to generate pseudo-labels for learning to rank.

related to the query topic, these queries created effective pseudo-labels through the iterative procedure of transductive learning. Table 2 shows the query types and their corresponding roles while selecting high-quality queries. To balance the distributions of high-quality and low-quality queries in the training/testing/validation sets, three-fold cross-validation experiments were conducted while selecting high quality queries. Note that during the process of building the ranker to predict retrieval performance in terms of mean average precision, each query-document pair was represented by predefined document features. When constructing the query quality estimator, each query-document pair was represented by the exploited quality-related features, namely the NQC-based features.

B. TAGGING UNLABELED DATA BY TRANSDUCTIVE LEARNING

It is common for documents relevant to different queries to have diverse distributions, and ignoring the quality of queries used for training may hurt the effectiveness of retrieval in learning to rank because it leads to the aggregation of low quality pseudo-labels during the iterations of transductive learning. After the selection of the high quality queries, all queries were divided into two classes, high quality and low quality queries. The former enabled learning to rank to improve its effectiveness of retrieval because they can generate pseudo-labels with a higher average quality. In contrast, the latter brought about only a minor improvement in performance, or can even degraded retrieval performance. Thus, the pseudo-labels created by these low quality queries were unreliable. In the proposed method, the 100 highest documents in the initial content-based retrieval were treated as unlabeled examples. For high quality queries, we extracted the highest and lowest ranked documents as input, and trained a learner by applying learning to rank algorithms. The remaining unlabeled documents of the top 100 documents were re-ranked by our learner, and we picked the most relevant and irrelevant examples from this list to update the training data. A new

learner was then retrained by the most recent training data to re-rank the remaining unlabeled examples once again. The procedure of training and selecting pseudo-labels was iterated until the halting criterion was met. The algorithm for tagging the unlabeled data is shown in Fig. 2. Finally, the pseudo-examples iteratively generated by the transductive learning algorithm were provided as input for learning to rank.

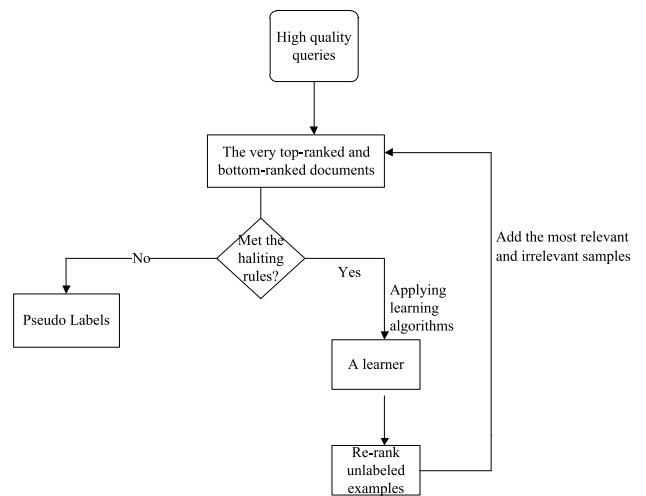


FIGURE 2. The procedure of transductive learning, in which only high-quality queries are used.

C. RETRIEVAL DOCUMENTS BY LEARNING TO RANK

To examine the retrieval performance of our proposed method, the pseudo-labels aggregated iteratively by it were used in learning to rank. We now introduced the query partitions in the learning to rank procedure. Four types of queries were used in our experiments: training, validation, target, and test queries. Both relevant and irrelevant examples were extracted from the documents retrieved by the training queries as initial pseudo-training data to train the ranking

model. The queries in the validation set were used to determine the settings of the tunable parameters for learning to rank. Thus the optimal parameters of the learning to rank algorithm were obtained and used for the target queries to iteratively generate the final pseudo-examples. Finally, the ranking model learned from the pseudo-training examples generated by the target queries was used to generate document rankings for each query in the test set.

In our experiments, the high quality queries selected by the proposed transductive learning method were randomly split into three parts: a training queries' set, a validation queries' set, and a test queries' set. The low quality queries were in turn arbitrarily split: the remaining part of the validation set, and the test queries. There was no intersection among the training, validation, and testing sets, but there was an overlap between the target and the test queries. While manually assigned labels were used for the validation set during the optimization of the parameters, there was no overlap between the validation query set and the training/testing query set. Such an experimental setup is common for semi-supervised learning to rank approaches, such as in [18], [40].

D. MACHINE LEARNING ALGORITHMS FOR LEARNING TO RANK AND CLASSIFICATION

In this section, we introduced the classification algorithms used to select high quality training queries, and the learning to rank algorithm to process the tagging of unlabeled data and searching the results of retrieval for a given query.

1) LEARNING TO RANK ALGORITHMS

Three major learning to rank algorithms, the pointwise, pairwise, and listwise approaches, are commonly used in information retrieval, and where the pairwise and listwise methods have been shown to outperform the pointwise approach in terms of retrieval performance [35]–[38]. In this paper, RankSVM [39], a classical pairwise approach, was chosen to train the ranking algorithms. The literature has shown the benefits of RankSVM in search tasks [39], [40].

Given a query Q and the set of documents D relevant to it, RankSVM transforms the ranking task into a classification task by leveraging the relative relevance of each document pair in D . As a classical pairwise learning to rank approach, RankSVM took a pair of documents in D as input, and its optimization was similar to that of SVM, where it added an SVM regularization for margin maximization to the objective and a parameter C that allowed a trading-off margin against training error:

$$\begin{aligned} \min_{w, \xi} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & \langle x_i^{(1)} - x_i^{(2)}, w \rangle \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \end{aligned} \quad (2)$$

where w is a weight vector adjusted during the learning procedure, x is a vector of documents, y is the relevance of a document to the given query, and ξ is a slack variable.

2) CLASSIFICATION ALGORITHMS

In information retrieval, in case labeled data are available, many classification algorithms, such as the neuron network, gradient boosting, logistic regression, and SVM, can be applied [41]. Because only a small amount of reliable data was available in our application, we used classification algorithms for predicting the retrieval-related benefits of each given query while estimating query quality. Both logistic regression and naive Bayes were used to classify the queries. We now briefly introduce the two classifications.

The idea of logistic regression was nearly identical to that of linear regression. However, logistic regression involves a step that uses the sigmoid function to convert the output of linear regression to return a probability value that can then be mapped to two or more discrete classes. The kernel of classification in logistic regression was that it utilized the sigmoid function. Similar to linear regression, it had a cost function obtained by minimizing the final solution [42]. When logistic regression was applied to binary classification, the parametric model was as follows:

$$\begin{aligned} P(Y = 0|X) &= \frac{\exp(w_0 + \sum_{i=1}^n w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^n w_i X_i)} \\ P(Y = 1|X) &= \frac{1}{1 + \exp(w_0 + \sum_{i=1}^n w_i X_i)} \end{aligned} \quad (3)$$

where w is the weight of each vector X .

Logistic regression achieved classification by learning hyper-planes. In contrast, naive Bayes carried out classification by taking the unique approach of considering the probabilities of features. The Bayesian method used knowledge of probability to classify the sample dataset. Because of its sound mathematical foundation, the false positive rate of the Bayesian classification algorithm was low. Its characteristic was to combine the prior probability with the posterior probability, which helped avoid subjective bias and the over-fitting phenomenon. The Bayesian classification algorithm yielded high accuracy when the dataset was large and the algorithm itself was relatively simple. Naive Bayes classifiers have worked well in many complex situations [43]–[45]:

$$P(C_k|x) = \frac{P(C_k) \prod_{i=1}^n P(x_i|C_k)}{\sum_{k=1}^K [P(C_k)] \prod_{i=1}^n P(x_i|C_k)} \quad (4)$$

where C_k is a class, x is a vector space, and x_i is the feature that is a part of x .

IV. EXPERIMENTAL SETUP

A. DATASET

To examine the effectiveness of retrieval of our proposed method, a series of extensive experiments were conducted on LETOR 4.0, which is a standard dataset for learning to rank [11]. LETOR 4.0 contains 25 million pages from the Gov2 dataset, and two query sets from Million Query track1 of TREC 2007 and TREC 2008. The two query sets are hereinafter denoted by MQ2007 and MQ2008, respectively. MQ2007 contained 1,692 queries with labeled documents

and MQ2008 contained 784 with labeled documents. In the experiments, we remove those queries having no relevant documents while evaluating the effectiveness of our proposed transductive learning approaches, as suggested in [19], [20]. Documents in LETOR 4.0 were represented by predefined features. Each row in the dataset was a query-document pair, in which the first column was the relevance label of the document to the given query, the second column was query id, and the subsequent columns were 46-dimensional feature vectors. In each query-document pair, the larger the relevance label was, the more relevant the document was to the given query [11].

In the tables provided below, a* indicates that a statistically significant difference between the baseline and our proposed method was observed. Considering that the T-test is commonly used as a standard statistical test for information retrieval on LETOR 4.0, our significance test was conducted using it at a 0.05 confidence level.

B. EXPERIMENTAL SETUP AND EVALUATION MEASURE

The aim of our experiments was to evaluate the effectiveness of the proposed transductive learning, in which pseudo-examples generated iteratively by utilizing high quality queries were used to learn a ranking model for learning to rank. Five-fold cross-validation experiments were conducted using our proposed method. To examine the retrieval performance of the proposed method, four strong baselines were used.

- Semi-supervised learning approaches. In this approach, as suggested in [4], [6], [8], all queries despite quality were utilized and the pseudo-examples were extracted from the documents ranked high and low. In traditional transductive learning, all queries were divided into three partitions: test queries, training queries, and validation queries.
- Clustered-based transductive learning proposed in [19]. In this study, the authors presented a semi-supervised learning algorithm that used the clustered-based transductive method combined with a non-measure-specific listwise approach for learning to rank.
- A training query-filtering approach for semi-supervised learning to rank, in [20]. In this method, queries were classified by utilizing only sparse labeled data for applications.
- The state-of-the-art deep neural ranking models, as proposed in [46], [47]. DeepRank is a deep learning architecture that models the relevance of a document for a given query by simulating the process of human judgment [46]. Another neural ranking model is HiNT, which is a hierarchical neural matching model used to capture diverse patterns of relevance in ad-hoc retrieval. In HiNT, a deep neural network is employed to support high-quality relevance signal generation and flexible relevance assessment strategies [47]. For DeepRank and HiNT, we used the codes released by the authors.

Standard evaluation measures in information retrieval such as the mean average precision (MAP) and average normalized discounted cumulative gain (AVG_NDCG) were used in our experiments. The MAP, as a standard evaluation measure for information retrieval, was computed using the mean of average precisions on all queries. However, precision is the fraction of the number of relevant documents in all documents for a given query. Thus, the MAP was given as mean of precisions all over queries. For the given queries, a high-MAP was obtained if the relevant documents of the queries were mostly ranked high. The MAP is given as follows:

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \quad (5)$$

where Q is the number of all queries available, $|Q|$ is the total number of queries, m_j is the number of relevant documents for a given query, and R_{jk} is the document set in the results ranked before the relevant document d_k in terms of ranking.

The normalized discounted cumulative gain (NDCG) is also an evaluation measure commonly used in information retrieval. It leverages the relevance judgment in terms of multiple order categories. It is given as:

$$NDCG(Q, K) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_{j,k} \sum_{m=1}^k \frac{2^{R(j,m)} - 1}{\log(1 + m)} \quad (6)$$

where Q is number of all queries available, $|Q|$ is the total number of queries, $R(j,m)$ is the relevance label for document m to the given query j , and $Z_{j,k}$ denotes the normalization factor. In our experiments, the mean NDCG (AVG_NDCG) was also used as evaluation measure.

V. RESULTS

A. PERFORMANCE COMPARISON BETWEEN PROPOSED TRANSDUCTIVE LEARNING AND TRADITIONAL TRANSDUCTIVE LEARNING

We first evaluated the effectiveness of retrieval of our proposed method (PTL) in comparison with traditional transductive learning (TTL), which used all available queries to generate pseudo-labels for learning to rank. In our proposed method, two classification algorithms were used, naive Bayes and logistic regression. In the details of the experiments provided below, the proposed methods with the two classifications are respectively denoted by PTL_NB and PTL_LR. Table 3 shows that a significant improvement over the baseline was obtained when the proposed method was applied to the MQ2007 and MQ2008.

An increase of 0.85% in terms of AVG_NDCG on the MQ2007 was obtained; although a small decrease in terms of MAP was noted when naive Bayes was used to classify the test queries. On the MQ2008, the proposed method, PTL, outperformed the baseline both in terms of MAP and AVG_NDCG. Improvements of 0.81% in terms of the MAP and 1.69% in terms of the AVG_NDCG were observed when PTL_NB was applied to the MQ2008. It also showed significant advantages over the conventional method on LETOR 4.0

TABLE 3. Comparison of performance between the proposed transductive learning (PTL) and traditional transductive learning (TTL).

MQ2007			
	TTL	PTL NB	PTL LR
MAP	0.5186	0.5172, - 0.27%	0.5262 , +1.47% *
AVG_NDCG	0.5532	0.5579 , +0.85% *	0.5614 , +1.48% *
MQ2008			
	TTL	PTL NB	PTL LR
MAP	0.6533	0.6586 , +0.81% *	0.6570, +0.57%
AVG_NDCG	0.6632	0.6744 , +1.69% *	0.6700 , +1.03% *

when logistic regression was used as the classification algorithm. For example, with MAP as evaluation metric the proposed method PTL_LR outperformed the baseline TTL by 1.47%. Furthermore, PTL_LR showed notable advantages in terms of AVG_NDCG. On the MQ2008, an improvement of 1.03% in terms of the AVG_NDCG was obtained. The results thus showed that the query quality had a significant impact on the quality of pseudo-labels in transductive learning, and that it was necessary to distinguish high quality queries from low quality ones.

B. COMPARISON BETWEEN PROPOSED TRANSDUCTIVE LEARNING AND CLUSTERED-BASED TRANSDUCTIVE LEARNING

In this section, we compared our proposed method (PTL) with the clustered-based transductive learning (CTL) proposed in [19]. As shown in Table 4, when naive Bayes was used to classify the queries, the proposed method PTL_NB outperformed CTL_LISTMLE by 7.77% and CTL_LambdaRank by 2.57% in terms of AVG_NDCG. However, no significant advantage was obtained in the other cases. The improvements on LETOR 4.0 were notable both on the MQ2007 and the MQ2008 in most cases when logistic regression was used to assess query quality. For example, in terms of the MAP, the proposed method PTL_LR outperformed the strong baselines significantly, by 0.82% compared with the CTL_LISTMLE and 1.42% with the CTL_LambdaRank. However, no notable improvement in terms of the MAP was obtained between PTL_LR and CTL_LambdaRank. The table shows that the proposed method PTL performed well on the MQ2008. By contrast, on the MQ2007, the effectiveness of PTL_LR was superior to that of obtained by PTL_NB. We think that for naive Bayes classification, the algorithm assumes that the features are independent of one another, but the document features in LETOR 4.0 are intimately related because features are scores obtained by applying different retrieval models. Thus, PTL_NB did not perform as well as PTL_LR. In case a few labeled examples were available, the proposed approach, PTL, instead of CTL was more suitable for creating pseudo-labels for learning to rank because the quality of the selected queries obtained is higher. Of course, for scenarios

involving no labeled data, the clustering-based transduction method is recommended for generating pseudo-labels for learning to rank.

C. COMPARISON BETWEEN PROPOSED TRANSDUCTIVE LEARNING AND CLASSIFICATION-BASED TRANSDUCTIVE LEARNING

Thirdly, we compared our proposed method with classification-based transductive learning as shown in Table 5. It shows that our method, PTL, outperformed the classification-based method in most cases, although the improvements were not notable on the MQ2008 when naive Bayes was applied to distinguish query quality. A significant advantage was obtained when its performance on the MQ2007 was evaluated in terms of AVG_NDCG. PTL_NB significantly outperformed the classification-based method STL (an improvement of 1.05%), with the exception that the retrieval performance of our proposed transductive learning, PTL, was notably worse on MQ2008 when logistic regression was used as classification algorithm and AVG_NDCG as evaluation measure. A significant decrease of 1.06% was observed by applying PTL_LR. In general, the proposed transductive learning PTL_NB performed well when naive Bayes was applied as classification method, whereas PTL_LR did not perform as well as expected. It is believed that the job of sampling most unconfident data nearby the classification is mainly around the classification boundaries, which may not bring benefit for logistic regression achieving classification by learning hyper-planes. However, the parameter optimization of naive Bayes is maximum likelihood estimation and the quantity of the pseudo-labeled data may increase the precision.

D. COMPARISON BETWEEN PROPOSED TRANSDUCTIVE LEARNING AND NEURAL RANKING MODELS

Finally, we compared our proposed method with the state-of-the-art neural ranking models shown in Table 6. Note that we conducted experiments only on MQ2007 because the number of queries in MQ2008 was too small for training the deep neural ranking models. The table shows that with MAP as evaluation measure, none of the deep neural ranking models outperformed the proposed transductive learning method. For example, PTL_LR was superior to the best deep neural ranking model, HiNT, on MQ2007 by 4.82%; while PTL_NB was superior to the best deep neural ranking model by 3.03% on MQ2007. The results show that in terms of information retrieval, although more complex models were trained using deep neural ranking, the automatically learned features for ranking may not be sufficient to improve the effectiveness of retrieval. Our PTL significantly outperformed the deep neural ranking models in terms of MAP, which shows that the improvement in the quality of the training data played an important role in enhancing the effectiveness of information retrieval.

TABLE 4. Comparison of performance between the proposed transductive learning (PTL) and clustered-based transductive learning (CTL).

MQ2007				
	CTL LISTMLE	CTL LambdaRank	PTL NB	PTL LR
MAP	0.5219	0.5188	0.5172,-0.90%,-0.31%	0.5262, +0.82% *, +1.42%*
AVG NDCG	0.5598	0.5545	0.5579,-0.34%,+0.61%	0.5614,+0.29%,+1.24%*
MQ2008				
	CTL LISTMLE	CTL LambdaRank	PTL NB	PTL LR
MAP	0.6538	0.6572	0.6586,+0.73%,+0.21%	0.6570,+0.49%,-0.03%
AVG NDCG	0.6258	0.6575	0.6744,+7.77%*+2.57%*	0.6700,+7.06%*,+1.90%*

TABLE 5. Comparison of performance between the proposed transductive learning (PTL) and classification-based transductive learning (STL).

MQ2007				
	STL NB	PTL NB	STL LR	PTL LR
MAP	0.5178	0.5172, -0.12%	0.5242	0.5262, +0.38%
AVG NDCG	0.5521	0.5579, +1.05% *	0.5587	0.5614, +0.48%
MQ2008				
	STL NB	PTL NB	STL LR	PTL LR
MAP	0.6578	0.6586, +0.12%	0.6619	0.6570, -0.74%
AVG NDCG	0.6741	0.6744, +0.04%	0.6772	0.6700, -1.06% *

TABLE 6. Comparison of performance between the proposed transductive learning (PTL) and deep neural ranking models.

DeepRank	PTL NB	PTL LR
0.4970	0.5172, +4.06% *	0.5262, +5.88% *
HiNT	PTL NB	PTL LR
0.5020	0.5172, +3.03% *	0.5262, +4.82% *

TABLE 7. The precision and recall of the selected queries on MQ2007 and MQ2008.

MQ2007			MQ2008		
Recall	Precision	Accuracy	Recall	Precision	Accuracy
55.32%	93.20%	98.60%	85.82%	51.23%	62.23%

VI. DISCUSSION

Table 7 gives the precision and recall of the selected high-quality queries measured against the ground truth. If a query yielded a positive benefit for the test queries during the assessment of the quality of query estimation, it was considered a high-quality query. Otherwise, it was considered a low quality one. On the MQ2007, a high precision of 93.2% was obtained in this way. Although a moderately low accuracy was obtained on the MQ2008, the high recall (85.82%) indicated that most high quality training queries were correctly identified by utilizing the proposed transductive learning method. Therefore, an effective and robust retrieval performance was still achieved by using the appropriate classification algorithm.

The above results showed that the PTL exhibited significant advantages in terms of retrieval performance compared with traditional transduction (TTL) and clustered-based transduction (CTL). Although enhancements in the effectiveness of retrieval were also noted, PTL was not always superior to STL. We thus investigated the effectiveness of classification-based transduction, STL, through iterative

tests. Fig. 3 shows us a comparison between PTL and STL in terms of the MAP at each iteration from 1 to 15 on the MQ2007 when logistic regression was used to estimate query quality. It is clear that PTL outperformed STL in most cases. Note also that it yielded the highest values in terms of the MAP in the third and fourth rounds.

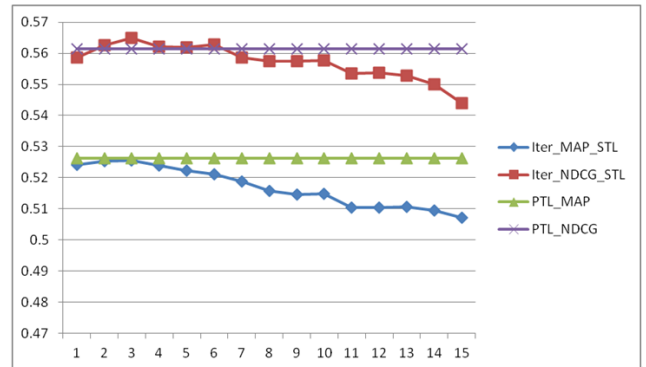


FIGURE 3. Comparison between PTL and STL in terms of the MAP in each iteration from 1 to 15 on MQ2007 when logistic regression was used for classification.

We also calculated the MAP and AVG_NDCG between PTL and STL in each iteration from 1 to 15 on the MQ2008 when logistic regression was used for classification. Although, PTL did not yielded as high an MAP as STL in the second round, a high retrieval performance was obtained when it was applied to the MQ2008. Fig. 3 and Fig. 4 show that regardless of the datasets, the retrieval performance of PTL and STL exhibited a downward trend with the number of iterations. A high value of the MAP was obtained around the second to fourth rounds, both for PTL and STL, which also suggests that the pseudo-labels in these rounds were more likely to supply useful information for semi-supervised learning when applied to ranking queries.

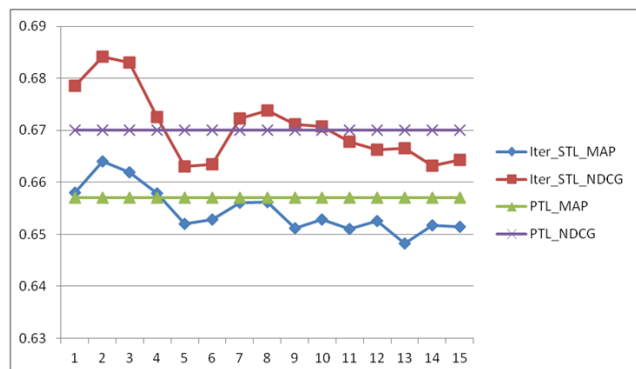


FIGURE 4. Comparison between PTL and STL in terms of the MAP of each iteration from 1 to 15 on the MQ2008 when logistic regression was used for classification.

VII. CONCLUSION

In this paper, we proposed a transductive learning approach that distinguished high quality queries from low quality ones to iteratively generate pseudo-labels for learning to rank. Specifically, to improve the quality of recognition of relevant and irrelevant pseudo-labels, we estimated the quality of each given query by judging whether it could enhance retrieval performance with a limited number of reliable labels along with data with low confidence close to the classification boundaries. By building a classifier between features that influence query quality and performance-related benefits, high quality queries were extracted by our proposed approach. The proposed transductive method was significantly different from traditional transduction in that only the selected queries were used to iteratively aggregate pseudo-labels for learning to rank. Experiments were conducted on the standard dataset LETOR 4.0, and the results showed that a statistically significant improvement over the baselines was achieved by applying our proposed transductive learning approaches in most of the cases considered. In addition, our study proved that with a limited number of labeled examples available, the proposed approach, PTL, was suitable for creating pseudo-labels for learning to rank because the quality of the selected queries in it can supply useful information.

REFERENCES

- [1] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng, "A deep look into neural ranking models for information retrieval," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102067.
- [2] Y. Liu, W. Zhang, and W. Du, "Progressive system: A deep-learning framework for real-time data in industrial production," *Processes*, vol. 8, no. 6, p. 649, May 2020.
- [3] T.-Y. Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retr.*, vol. 3, no. 3, pp. 225–331, 2009.
- [4] M. Szummer and E. Yilmaz, "Semi-supervised learning to rank with preference regularization," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2011, pp. 269–278.
- [5] V. N. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley-Interscience, 1998.
- [6] R. El-Yaniv and D. Pechyony, "Stable transductive learning," in *Proc. 19th Annu. Conf. Learn. Theory*, 2006, pp. 35–49.

- [7] X. Wang, J. Wen, S. Alam, Z. Jiang, and Y. Wu, "Semi-supervised learning combining transductive support vector machine with active learning," *Neurocomputing*, vol. 173, pp. 1288–1298, Jan. 2016.
- [8] J. X. Huang, J. Miao, and B. He, "High performance query expansion using adaptive co-training," *Inf. Process. Manage.*, vol. 49, no. 2, pp. 441–453, Mar. 2013.
- [9] T. Reitmaier, A. Calma, and B. Sick, "Transductive active learning—a new semi-supervised learning approach based on iteratively refined generative models to capture structure in data," *Inf. Sci.*, vol. 293, pp. 275–298, Feb. 2015.
- [10] S. Karlos, V. G. Kanas, C. Aridas, N. Fazakis, and S. Kotsiantis, "Combining active learning with self-train algorithm for classification of multi-modal problems," in *Proc. 10th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2019, pp. 1–8.
- [11] T. Qin and T. Liu. (2013). *Introducing Letor 4.0 Datasets*. Microsoft Research Asia. [Online]. Available: <https://www.microsoft.com/en-us/research/project/letor-learning-rank-information-retrieval/>
- [12] Z. Xie, Y. Xu, and Q. Hu, "Uncertain data classification with additive kernel support vector machine," *Data Knowl. Eng.*, vol. 117, pp. 87–97, Sep. 2018.
- [13] L. Zhang, S. Valae, Y. Xu, L. Ma, and F. Vedadi, "Graph-based semi-supervised learning for indoor localization using crowdsourced data," *Appl. Sci.*, vol. 7, no. 5, p. 467, Apr. 2017.
- [14] M. A. Humayun, I. Hameed, S. M. Shah, S. Hassan Khan, I. Zafar, S. B. Ahmed, and J. Shuja, "Regularized Urdu speech recognition with semi-supervised deep learning," *Appl. Sci.*, vol. 9, no. 9, p. 1956, May 2019.
- [15] Z. Ji, Y. Pang, Y. He, and H. Zhang, "Semi-supervised LPP algorithms for learning-to-rank-based visual search reranking," *Inf. Sci.*, vol. 302, pp. 83–93, May 2015.
- [16] G. Franzese, N. Linty, and F. Doves, "Semi-supervised GNSS scintillations detection based on deepinforax," *Appl. Sci.*, vol. 10, no. 1, p. 381, 2020.
- [17] M. Stikic, K. V. Laerhoven, and B. Schiele, "Exploring semi-supervised and active learning for activity recognition," in *Proc. 12th IEEE Int. Symp. Wearable Comput.*, Sep./Oct. 2008, pp. 81–88.
- [18] X. Huang, Y. Huang, M. Wen, A. An, Y. Liu, and J. Poon, "Applying data mining to pseudo-relevance feedback for high performance text retrieval," in *Proc. 6th Int. Conf. Data Mining (ICDM)*, Dec. 2006, pp. 295–306.
- [19] A. U. Rahangdale and S. Raut, "Clustering-based transductive semi-supervised learning for learning-to-rank," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 33, no. 12, 2019, Art. no. 1951007.
- [20] X. Zhang, B. He, and T. Luo, "Training query filtering for semi-supervised learning to rank with pseudo labels," *World Wide Web*, vol. 19, no. 5, pp. 833–864, Sep. 2016.
- [21] K. Muandet, S. Marukat, and C. Nattee, "Query selection via weighted entropy in graph-based semi-supervised classification," in *Proc. Asian Conf. Mach. Learn.*, 2009, pp. 278–292.
- [22] X. Geng, T. Qin, T.-Y. Liu, X.-Q. Cheng, and H. Li, "Selecting optimal training data for learning to rank," *Inf. Process. Manage.*, vol. 47, no. 5, pp. 730–741, Sep. 2011.
- [23] X. Zhang, Z. Q. Zhao, C. H. Liu, C. Zhang, and Z. Cheng, "Semi-supervised learning to rank with uncertain data," in *Proc. 16th Web Inf. Syst. Appl. (WISA)*, 2019.
- [24] J. Xu, C. Chen, G. Xu, H. Li, and E. R. T. Abib, "Improving quality of training data for learning to rank using click-through data," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining (WSDM)*, 2010, pp. 171–180.
- [25] V. R. Carvalho, J. L. Elsas, W. W. Cohen, and J. G. Carbonell, "A meta-learning approach for robust rank learning," in *Proc. SIGIR Workshop Learn. Rank Inf. Retr.*, 2008, pp. 1–9.
- [26] X. Geng, T. Liu, T. Qin, and H. Li, "Feature selection for ranking," in *Proc. SIGIR Conf. Res. Develop. Inf. Retr.*, 2007, pp. 407–414.
- [27] X. Geng, T. Qin, T.-Y. Liu, and X.-Q. Cheng, "A noise-tolerant graphical model for ranking," *Inf. Process. Manage.*, vol. 48, no. 2, pp. 374–383, Mar. 2012.
- [28] P. K. Mallapragada, R. Jin, and A. K. Jain, "Active query selection for semi-supervised clustering," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
- [29] Q. Li, X.-M. Wu, H. Liu, X. Zhang, and Z. Guan, "Label efficient semi-supervised learning via graph filtering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9582–9591.
- [30] M. Hasanain and T. Elsayed, "Query performance prediction for microblog search," *Inf. Process. Manage.*, vol. 53, no. 6, pp. 1320–1341, Nov. 2017.

- [31] S. Cronen-Townsend, Y. Zhou, and W. B. Croft, "Predicting query performance," in *Proc. 25th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2002, pp. 299–306.
- [32] Y. Zhou and W. B. Croft, "Query performance prediction in Web search environments," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2007, pp. 543–550.
- [33] A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits, "Predicting query performance by query-drift estimation," *ACM Trans. Inf. Syst.*, vol. 30, no. 2, pp. 1–15, 2012.
- [34] X. Zhang, B. He, T. Luo, and B. Li, "Query-biased learning to rank for real-time Twitter search," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2012, pp. 1915–1919.
- [35] S. Chelaru, C. Orellana-Rodriguez, and I. S. Altingovde, "How useful is social feedback for learning to rank YouTube videos?" *World Wide Web*, vol. 17, no. 5, pp. 997–1025, Sep. 2014.
- [36] Y. Ganjisaffar, R. Caruana, and C. V. Lopes, "Bagging gradient-boosted trees for high precision, low variance ranking models," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. (SIGIR)*, 2011, pp. 85–94.
- [37] Z. Luo, M. Osborne, and T. Wang, "An effective approach to tweets opinion retrieval," *World Wide Web*, vol. 18, no. 3, pp. 545–566, 2013.
- [38] N. Collier, A. Oellrich, and T. Groza, "Concept selection for phenotypes and diseases using learn to rank," *J. Biomed. Semantics*, vol. 6, no. 1, p. 24, Dec. 2015.
- [39] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2002, pp. 133–142.
- [40] X. Zhang, B. He, T. Luo, D. Li, and J. Xu, "Clustering-based transduction for learning a ranking model with limited human labels," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. CIKM*, 2013, pp. 1777–1782.
- [41] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
- [42] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.
- [43] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Mach. Learn.*, vol. 29, nos. 2–3, pp. 103–130, 1997.
- [44] C. Xi-cheng, B.-B. Fan, W.-J. Yang, G.-L. Tian, and K. Wu, "Air-attack weapon identification model of weighted Navie Bayes based on SOA," *J. Phys., Conf. Ser.*, vol. 1060, Jul. 2018, Art. no. 012054.
- [45] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019.
- [46] L. Pang, Y. Lan, J. Guo, J. Xu, J. Xu, and X. Cheng, "DeepRank: A new deep architecture for relevance ranking in information retrieval," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 257–266.
- [47] Y. Fan, J. Guo, Y. Lan, J. Xu, C. Zhai, and X. Cheng, "Modeling diverse relevance patterns in ad-hoc retrieval," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 375–384.



XIN ZHANG received the bachelor's degree from Shandong University and the Ph.D. degree from the University of Chinese Academy of Sciences, China. She is currently a Lecturer with the Department of Artificial Intelligence and Big Data, Hefei University. Her research interests include big data information processing and information retrieval.



ZHI CHENG received the M.S. degree from the Chinese Academy of Science, in 2013, and the Ph.D. degree from the University of Science and Technology of China, in 2017. She is currently a Lecturer with the Department of Artificial Intelligence and Big Data, Hefei University. Her research interests include image processing and computer version.

...