Ranking Experts with Discriminative Probabilistic Models

Yi Fang
Department of Computer Science
Purdue University
250 N. University Street
West Lafayette, IN 47907
fangy@cs.purdue.edu

Luo Si
Department of Computer Science
Purdue University
250 N. University Street
West Lafayette, IN 47907
lsi@cs.purdue.edu

Aditya P. Mathur
Department of Computer Science
Purdue University
250 N. University Street
West Lafayette, IN 47907
apm@cs.purdue.edu

ABSTRACT

In the realistic settings of expert finding, the evidence for expertise often comes from heterogeneous knowledge sources. As some sources tend to be more reliable and indicative than the others, different data sources need to receive different weights to reflect their degrees of importance. However, most previous studies in expert finding did not differentiate data sources, which may lead to unsatisfactory performance in the settings where the heterogeneity of data sources is present.

In this paper, we investigate how to merge and weight heterogeneous knowledge sources in the context of expert finding. A relevance-based supervised learning framework is presented to learn the combination weights from training data. Beyond just learning a fixed combination strategy for all the queries and experts, we propose a series of probabilistic models which have increasing capability to associate the combination weights with specific experts and queries. In the last (and also the most sophisticated) proposed model, the combination weights depend on both expert classes and query topics, and these classes and topics are derived from expert and query features. Compared with expert and query independent combination methods, the proposed combination strategy can better adjust to different types of experts and queries. In consequence, the model yields much flexibility of combining data sources when dealing with a broad range of expertise areas and a large variation in experts. Empirical studies on a real world faculty expertise testbed demonstrate the effectiveness and robustness of the proposed learning based models.

Categories and Subject Descriptors

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

General Terms

Design, Algorithms, Experimentation

Keywords

Expert finding, Learning to rank

1. INTRODUCTION

With vast amount of information available in large organizations, there are increasing needs for users to find not only documents, but also people who have specific knowledge in a required area. For example, many companies can deliver efficient customer services if the customer complaints can be directed to the appropriate staff. Similarly, conference organizers need to locate the program committee members based on their research expertise to assign submissions. Academic institutions want to publicize their faculty expertise to funding agencies, industry sponsors, and potential research collaborators. Students are also avid seekers for prospective advisers with matched research interests. Thus, finding the right person in an organization with the appropriate expertise is often crucial in many enterprise applications.

The expert finding task is generally defined as follows: given a keyword query, a list of experts and a collection of supporting documents, rank those experts based on the information from the data collection. Expert finding is similar to the traditional ad-hoc retrieval task since both tasks are targeted to find relevant information items given a user query. The major difference is that in the realistic settings of expert finding, the supporting evidence for expertise usually comes from a wide range of heterogeneous data sources such as research homepages, technical reports, publications, projects, course descriptions, and email discussions. However, most previous studies did not differentiate data sources and consequently how to merge and weight these heterogeneous sources in the context of expert finding has not been fully investigated.

In this paper, we present three discriminative probabilistic models for ranking experts by learning the combination weights of multiple data sources. The first model can be regarded as an application of logistic regression to ranking experts, which serves as the basis of the other more advanced models. The other two proposed models consider the latent class variables underlying the observed experts or/and queries. In the latent expert and query topic model that we proposed, the combination weights depend on both expert classes and query topics. In consequence, the weights can be better adjusted according to what characteristics the experts have and what types of information needs users express in the queries. The model offers probabilistic semantics for the latent expert/query topics and thus allows mixing multiple expert and query types for a single expert and query.

Although many query dependent resource merging methods have been proposed (for other IR tasks), to the best of our knowledge, there is no prior work on modeling the dependencies of the combination strategy on both queries and searched entities (e.g., documents or experts). In particular, the dependency on the searched experts is prominent in the scenario of expert finding. In the experiments, the proposed discriminative models have shown to have better performance than the prior solutions on a real world faculty expertise testbed (i.e., the Indiana Database of University Research Expertise (INDURE)[1]). Different versions of the models with different types of features are also compared. In addition, we have shown the robustness of the latent expert and query topic model by evaluating it with different document retrieval methods.

The next section discusses the related work. Section 3 describes the discriminative probabilistic models that discern the sources of expertise evidence. Section 4 explains our experimental methodology. Section 5 presents the experimental results and the corresponding discussions. Section 6 concludes.

2. RELATED WORK

Initial approaches to expert finding employed a manually constructed database which listed experts by category and subcategory [12]. These systems (often in the form of yellow pages) require a lot of manual work to classify expert profiles. More recent techniques locate expertise in an automatic fashion, but only focus on specific document types such as software [24] and email [6]. With abundant information becoming available on the Web, there is increasing interest in utilizing varied and heterogeneous sources of expertise evidence [2]. One early example is the P@noptic system [11], which builds a representation of each expert by concatenating all the documents associated with that expert. The user query is matched against this representation and thus finding experts is equally to retrieve documents. [23] treated the problem of ranking experts as a voting problem and explored 11 different voting strategies to aggregate over the documents associated to the expert. [2, 3], [14] and [31] proposed more formal methods for expert finding using language modeling. However, these methods do not differentiate document types, which may cause unsatisfactory performance in real world applications where some data sources are likely more reliable and indicative than others. Expert finding has attracted a lot of interest since the launch of Enterprise Track [10] at TREC and rapid progress has been made in modeling, algorithms, and evaluations. Nearly all the assumptions were made based on the characteristics of the W3C collection [9]. While in W3C the authors of documents are ambiguous, in more realistic settings it is reasonable to assume that the document-expert association is clear [3].

The voting process proposed for the expert finding task is also closely related to data fusion in metasearch [1] and collection fusion problem in distributed information retrieval [5]. The general retrieval source combination problem has been examined by a significant body of previous work. [15]'s method ranked documents based on the min, max, median, or sum of each document's normalized relevance scores over a set of systems. Linear combination and logistic regression models are explored by [30, 34, 33] in the context of data fusion. Although good results are achieved in specific cases, these techniques have not yet been shown to produce reliable improvement, which may come from the fact that their combination strategy keep constant for different query topics. Recent work [20] has led to query dependent combination methods, which project the query to the latent query topic space and learn the combination weights for each query topic from training data. In multimedia retrieval applications, the query dependent combination methods [21, 37] have been shown superior to query-independent combination. The work that is more closely related to ours is the work done by [36]. However, the prior work does not consider the dependency of the combination strategy on the searched entities (e.g., experts). In particular, this dependency is prominent in the case of expert finding. For example, some senior faculty do not have homepages and some junior faculty do not have supervised PhD dissertations. Thus, for senior faculty we may want to put less weight on homepages and similarly for junior faculty we expect less weight on dissertations.

On the other hand, our approach to expert finding also fits the paradigm of learning to rank, which is to construct a model or a function for ranking entities. Learning to rank has been drawing broad attention in the information retrieval community recently because many IR tasks are naturally ranking problems. Benchmark data sets such as LETOR [22] are also available for research on learning to rank. There are two general directions to rank learning. One is to formulate it into an ordinal regression problem by mapping the labels to an ordered set of numerical ranks [18, 8]. Another direction is to take object pairs as instances, formulate the learning task as classification of object pairs into two categories (correctly and incorrectly ranked), and train classification models for ranking [16, 19, 4, 17, 35]. More recently, the listwise approach, *ListNet* [7], is proposed to minimize a probabilistic listwise loss function instead of learning by minimizing a document pair loss functions. These methods are built on a solid foundation because it has been shown that they are closely related to optimizing the commonly used ranking criteria [28]. Although valuable work has been done for learning to rank for ad-hoc retrieval, very limited research has been conducted for designing learning models for ranking experts, which are generally associated with information from heterogeneous information sources.

3. DISCRIMINATIVE PROBABILISTIC MODELS FOR EXPERT FINDING

3.1 Notations and terminologies

Our approach to expert finding assumes that we have a heterogeneous document repository containing a set of documents from a mixture of K different knowledge sources. In the INDURE faculty expertise testbed, there exist four document sources, which are homepages, publications/supervised PhD dissertations, National Science Foundation (NSF) funding projects and general faculty profiles such as research keywords and affiliations. For the document collection, there are totally M experts and the document-expert association is clear (e.g., the authors of publications, the owners of homepages and the principal investigators of NSF projects). Within a single document source, each expert has a set of

[1]https://www.indure.org/

supporting documents and each document is associated with at least one expert. For a given query q and an expert e, we can obtain a ranking score, denoted by $s_i(e,q)$, from the i^{th} document source. In other words, $s_i(e,q)$ is the single-source ranking score for the expert e with respect to the query q. It is calculated by summing over the retrieval scores of the expert's supporting documents d in the single data source (i.e., $s_i(e,q) = \sum_{d \in F_i(e)} s_i(d,q)$ where $F_i(e)$ is the set of supporting documents for e in the i^{th} source, and more details are discussed in Section 4.1). $s_i(d,q)$ is the retrieval score for a single document d and can be calculated by any document retrieval model such as BM25 or language modeling. Obviously, if there is no document retrieved for e, $s_i(e,q)$ is equal to 0. Our goal is to combine $s_i(e,q)$ from K data sources to generate a final ranked list of experts.

3.2 Relevance based discriminative combination framework

Our basic retrieval models cast expert finding into a binary classification problem that treats the relevant query-expert pairs as positive data and irrelevant pairs as negative data. There exist many classification techniques in the literature and they generally fall into two categories: generative models and discriminative models. Discriminative models have attractive theoretical properties [26] and they have demonstrated their applicability in the field of IR. In presence of heterogeneous features due to multiple retrieval sources, the discriminative models generally perform better than their generative counterparts [25]. Thus, we adopt discriminative probabilistic models to combine multiple types of expertise evidence. Instead of doing a hard classification, we can estimate and rank the conditional probability of relevance with respect to the query and expert pair. Formally, given a query q and an expert e, we denote the conditional probability of relevance as $P(r|e,q)$ where $r \in \{1,-1\}$ indicating whether the expert e is relevant to the query q or not. The parametric form of the relevance probability can be expressed as follows in terms of logistic functions

$$P(r=1|e,q) = \sigma(\sum_{i=1}^{K} \omega_i s_i(e,q)) \qquad (1)$$

where $\sigma(x) = 1/(1+\exp(-x))$ is the standard logistic function and ω_i is the combination parameter for the i^{th} data source. This model is also known as logistic regression in which the parameters can be estimated by Newton's method. The experts are then ranked according to the descending order of $P(r=1|e,q)$.

3.3 Expert dependent probabilistic models

The model introduced in the last section provides a discriminative learning framework to estimate combination weights of multiple types of expertise evidence. In the model, the same combination weights are used for every expert to optimize the average performance. However, the best combination strategy for a given expert is not necessarily the best combination strategy for other experts. For example, many senior faculty members do not have homepages although they are probably very accomplished researchers in certain areas. On the other hand, new faculty members usually do not have any supervised PhD dissertations and thus it is not fair to put the same weights on dissertations as for senior faculty. In addition, many faculty members in the biology department do not have homepages to show their work in bioinformatics while most faculty in computer science in this area do have homepages. It will lead to unsatisfactory performance if we choose the same set of combination weights for all the experts regardless of their characteristics. Furthermore, real world expertise databases usually have data source missing problems. For example, some experts may have their homepages, but for some reason they are missing in the expertise database (e.g., homepage detection algorithms cannot perfectly discover all the homepages). It is not fair for these experts to be applied the same combination strategy as those experts with complete information. Therefore, we could benefit from developing an expert dependent model in which we can choose the combination strategy individually for each expert to optimize the performance for specific experts. Because it is not realistic to determine the proper combination strategy for every expert, we need to classify experts into one of several classes. The combination strategy is then tuned to optimize average performance for experts within the same class. Each expert within the same class shares the same strategy, and different classes of experts could have different strategies.

We present a latent expert class model (LEC) by introducing an intermediate latent class layer to capture the expert class information. Specifically, we can use a multinomial variable $z \in N$ to indicate which expert class the combination weights $\omega_{z\cdot} = (\omega_{z1}, ..., \omega_{zK})$ are drawn from. The choice of z depends on the expert e. The joint probability of relevance r and the latent variable z is given by

$$P(r=1,z|q,e;\alpha,\omega) = P(z|e;\alpha)P(r=1|q,e,z;\omega) \qquad (2)$$

where $P(z|e;\alpha)$ denotes the mixing coefficient which is the probability of choosing hidden expert classes z given expert q and α is the corresponding parameter. $P(r=1|q,e,z;\omega)$ denotes the mixture component which is a single logistic function in our case. $\omega = \{\omega_{zi}\}$ is the set of combination parameters where ω_{zi} is the weight for s_i under the class z. By marginalizing out the hidden variable z, the corresponding mixture model can be written as

$$P(r=1|q,e;\alpha,\omega) = \sum_{z=1}^{N_z} P(z|e;\alpha)\sigma(\sum_{i=1}^{K} \omega_{zi} s_i(e,q)) \qquad (3)$$

where N_z is the number of latent expert classes. If $P(z|e;\alpha)$ sticks to the multinomial distribution, the model cannot easily generalize the combination weights to unseen experts beyond the training collection, because each parameter in multinomial distribution specifically corresponds to a training expert. To address this problem, the mixing proportions $P(z|e;\alpha)$ can be modeled by a soft-max function $\frac{1}{Z_e}\exp(\sum_{j=1}^{L_z} \alpha_{zj} e_j)$ where Z is the normalization factor that scales the exponential function to be a proper probability distribution (i.e., $Z_e = \sum_z \exp(\sum_{j=1}^{L_z} \alpha_{zj} e_j)$). In this representation, each expert e is denoted by a bag of expert features $(e_1, ...e_{L_z})$ where L_z is the number of expert features. By plugging the soft-max function into Eqn. (3), we can get

$$P(r=1|q,e;\alpha,\omega) = \frac{1}{Z_e}\sum_{z}^{N_z} \exp(\sum_{j=1}^{L_z} \alpha_{zj} e_j)\sigma(\sum_{i=1}^{K} \omega_{zi} s_i(e,q))$$

Because α_{zj} is associated with each expert feature instead of

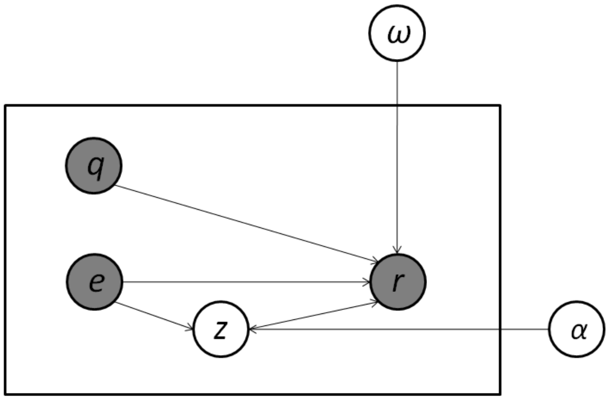

Figure 1: Graphical model representation for the LEC model. The nodes with known values are shaded and others are unshaded

each training expert, the above model allows the estimated α_{zj} to be applied in any unseen expert. The graphical representation for this model is shown in Figure 1.

3.3.1 Parameter estimation

The parameters can be determined by maximizing the following data log-likelihood function,

$$l(\omega,\alpha)=\sum_{u=1}^{N}\sum_{v=1}^{M}\log\Big(\textstyle\sum_z\big(\frac{1}{Z_{e_v}}\exp(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj})\big)\;\sigma\big(r_{uv}\sum_{i=1}^{K}\omega_{zi}s_i(e_v,q_u)\big)\Big) \quad (4)$$

where N is the number of queries, e_{vj} denotes the j^{th} feature for the v^{th} expert e_v and r_{uv} denotes the relevance judgement for the pair of (q_u, e_v). A typical approach to maximizing Eqn. (4) is to use the Expectation-Maximization (EM) algorithm [13], which can obtain a local optimum of log-likelihood by iterating E-step and M-step until convergence. The E-step can be derived as follows by computing the posterior probability of z given expert e_v and query q_u,

$$P(z|e_v,q_u)=\frac{\exp(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj})\sigma\big(r_{uv}\sum_{i=1}^{K}\omega_{zi}s_i(e_v,q_u)\big)}{\sum_z\exp(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj})\sigma\big(r_{uv}\sum_{i=1}^{K}\omega_{zi}s_i(e_v,q_u)\big)}$$

By optimizing the auxiliary Q-function, we can derive the following M-step update rules,

$$\omega_{z\cdot}^{*}=\arg\max_{\omega_{z\cdot}}\sum_{uv}P(z|e_v,q_u)\log\Big(\sigma\big(\sum_{i=1}^{K}\omega_{zi}s_i(e_v,q_u)\big)\Big)$$

$$\alpha_{z\cdot}^{*}==\arg\max_{\alpha_{z\cdot}}\sum_{u}\big(\sum_{v}P(z|e_v,q_u)\big)\log\Big(\frac{1}{Z_{e_v}}\exp\big(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj}\big)\Big)$$

The M-step can be optimized by any gradient descent method. In particular, we use Quasi-Newton method. When the log-likelihood converges to a local optimum, the estimated parameters can be plugged back into the model to compute the probability of relevance for unseen query and expert pairs. LEC can exploit the following advantages over the expert independent combination methods: 1) the combination parameters are able to change across various experts and hence lead to a gain of flexibility; 2) it offers probabilistic semantics for the latent expert classes and thus each expert can be associated with multiple classes; and 3) it can address the data source missing problem in a principled probabilistic framework.

3.4 Expert and query dependent probabilistic models

With the similar rationale to the expert dependent probabilistic model, the combination weights should also depend on specific queries. For example, for the query "history", we would like to have less weights put on NSF because the occurrence of "history" in NSF project descriptions is not likely to relate to the discipline in liberal arts, but more often to refer to the history of some technologies. Therefore, we should use different strategies to assign the combination weights for the queries coming from different topics. Based on the dependence of the combination strategy on both experts and queries, we propose the latent expert and query topic model (LEQT). The weight ω_{zti} now depends on both expert class z and query topic t. Assuming z and t are independent with each other giving e and q, the joint probability of relevance r and the latent variables (z, t) is,

$$P(r,z,t|q,e)=P(t|q)P(z|e)P(r|q,e,z,t) \quad (5)$$

By marginalizing out the hidden variables z and t, the corresponding mixture model can be written as

$$P(r=1|q,e)=\sum_{t=1}^{N_t}\sum_{z=1}^{N_z}P(t|q)P(z|e)\sigma\big(\sum_{i=1}^{K}\omega_{zti}s_i(e,q)\big) \quad (6)$$

The graphical representation for this learning model is shown in Figure 2. By plugging the soft-max functions for $P(z|e;\alpha)$ and $P(t|q;\beta)$, Eqn. (6) can then be reformulated as

$$P(r=1|q,e)=\frac{1}{Z_eT_q}\sum_{t=1}^{N_t}\sum_{z=1}^{N_z}\exp\big(\sum_{j=1}^{L_z}\alpha_{zj}e_j\big)\exp\big(\sum_{g=1}^{L_t}\beta_{tg}q_g\big)\sigma\big(\sum_{i=1}^{K}\omega_{zti}s_i(e,q)\big)$$

When $N_t = 1$, LEQT degenerates to LEC. When both numbers are equal to 1, LEQT becomes the logistic regression model in Section 3.2 (which is called expert and query independent (EQInd) model). Therefore, LEC and EQInd are all the special cases of LEQT.

For the LEQT model, the EM algorithm can be derived similarly. The E-step computes the posterior probability of the latent variables (z, t) given e and q as follows,

$$P(z,t|e_v,q_u)=\frac{\exp(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj})\exp(\sum_{g=1}^{L_t}\beta_{tg}q_{ug})\sigma(r_{uv}\sum_{i=1}^{K}\omega_{zti}s_i(e_v,q_u))}{\sum_{zt}\exp(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj})\exp(\sum_{g=1}^{L_t}\beta_{tg}q_{ug})\sigma(r_{uv}\sum_{i=1}^{K}\omega_{zti}s_i(e_v,q_u))}$$

In the M-step, we have the following update rule

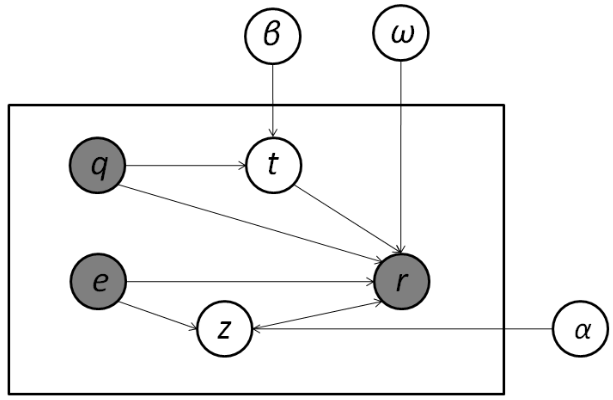

Figure 2: Graphical model representation for the LEQT model. The nodes with known values are shaded and others are unshaded

$$\omega^*_{zt\cdot} = \arg\max_{\omega_{zt\cdot}} \sum_{uv} P(z,t|e_v,q_u) \log\Big(\sigma\big(\sum_{i=1}^{K} \omega_{zti} s_i(e_v,q_u)\big)\Big)$$

$$\alpha^*_{z\cdot} = \arg\max_{\alpha_{z\cdot}} \sum_{v} \big(\sum_{ut} P(z,t|e_v,q_u)\big) \log\Big(\frac{1}{Z_{e_v}} \exp\big(\sum_{j=1}^{L_z} \alpha_{zj} e_{vj}\big)\Big)$$

$$\beta^*_{t\cdot} = \arg\max_{\beta_{t\cdot}} \sum_{u} \big(\sum_{vz} P(z,t|e_v,q_u)\big) \log\Big(\frac{1}{T_{q_u}} \exp\big(\sum_{g=1}^{L_t} \beta_{tg} q_{ug}\big)\Big)$$

3.5 Feature selection

To define the proposed models, we need to design a set of informative features for experts and queries. There are two useful principles to guide the design of suitable features: 1) they should be able to be automatically generated from expert and query descriptions, and 2) they should be indicative to estimate which latent classes the query or expert belongs to. In the case of academic expert finding, property based features can be used to investigate different characteristics of experts, which enable more appropriate usage of expertise information from different sources. Binary property features can be included to indicate whether information from different sources is available for a specific expert. For example, one feature will indicate whether the expert has a homepage and another feature will indicate whether the expert has any NSF project. These features will enable expert finding algorithms to shift their focus away from unavailable information sources by assigning appropriate weights. Numerical property features can also be utilized. For example, how long (in linear scale or in logarithmic scale) is a document from a particular information source such as length in the number of words or normalized length with respect to all documents from the same source. In addition, content based features can be used to investigate topic representation within documents from heterogeneous information sources and user queries, which enable better matching between expertise information in different sources and user queries. The content features can be represented as normalized weights for a set of topics (i.e., a multinomial distribution).

4. EXPERIMENTAL METHODOLOGY

4.1 The INDURE faculty expertise collection

The INDURE faculty expertise collection used in the experiments is constructed from the Indiana Database of University Research Expertise (INDURE) system developed at Purdue University. The INDURE effort aims at creating a comprehensive online database of all faculty researchers at academic institutions in the state of Indiana. Four universities currently participate in the project including Ball State University, Indiana University, Purdue University and University of Notre Dame. Together these universities involve over 12,000 faculty and research staff. The participating institutions are encouraged to log into the database to submit the basic information of their faculty such as college, department and research areas. The data in INDURE come from 4 different data sources: 1) the profiles filled out by individual faculty members and/or their department heads; 2) faculty homepages; 3) NSF funding project descriptions; 4) faculty publications and supervised PhD dissertations. The profiles include faculty research areas, which could be keywords from a predefined taxonomy[2] or free keywords that adequately describe the expertise. In this data collection, document-author associations are clear and the data is structured and clean. The collection covers a broad range of expertise areas, as one can typically find on intranets of universities.

In the INDURE faculty expertise data, some faculty have far more supervised PhD dissertations or NSF funded projects than others have. If we sum over all the supporting documents to calculate the single-source relevance score $s_i(e,q)$, it is possible that too many irrelevant documents are counted to exaggerate the final score. Therefore, in our experiments, we only consider the top scored supporting documents in an attempt to avoid the effect of small evidence accumulation (i.e., $s_i(e,q) = \sum_{d \in top(e,k)} s_i(d,q)$,where $top(e,k)$ denotes the set of top-k scored documents for e). To train the proposed models, 6,482 relevance judgments with 50 queries were made as training data. To evaluate the models, 50 test queries were submitted against the proposed models and the top 20 results returned by the algorithms for each test query were examined. Evaluation measures used were precision@5, @10, @15 and @20. Table 1 includes a subset of queries used in the evaluation.

Table 1: A subset of queries with relevance judgments used for evaluation

Information retrieval	Programming languages
Computational biology	Software engineering
Language education	Political economy
Mathematics education	Agricultural economics
Supply chain management	Developmental biology
Database	Asian history and civilizations

We apply the Indri retrieval model [32] as the default document retrieval method to obtain the single source retrieval score $s_i(d,q)$. The Indri toolbox[3] is used in the experiments. 21 query features and 34 expert features are chosen for the

[2]https://www.indure.org/hierarchy.cfm
[3]http://www.lemurproject.org/indri/

Table 2: Four types of features used in the experiments by the proposed models;

Source indicator	Whether certain data sources are absent for the given expert
Statistics	Lengthes and variance of the supporting documents in each data source; Number of words in the query; Number of associated NSF projects; etc
Category	Posterior probabilities of the expert and query belonging to the predefined classes
Others	Nubmer of images in the homepage; etc

proposed discriminative learning framework. As presented in Table 2, the total features can be divided into four sets: 1) source indicators that show whether certain data sources are absent for the given expert (F1); 2) query and document statistics (F2); 3) category features that indicate what categories the query or supporting documents belong to (F3); 4) other features such as the number of images in the homepages. The sizes of the corresponding feature sets are 4, 25, 16, and 10, respectively. The category features are obtained by calculating the posterior probabilities of the expert and query belonging to predefined categories. Eight categories such as Computer Science, Economy and Biology are chosen with a set of documents labeled for each category. Since the focus of this study is on the probabilistic models rather than feature engineering, we do not intend to choose a comprehensive set of features.

5. RESULTS AND DISCUSSIONS

An extensive set of experiments were designed on the INDURE faculty expertise testbed to address the following questions of the proposed research:

1) How good is the proposed discriminative probabilistic models compared with alternative solutions? We compare the results of the proposed methods with the results from prior solutions.

2) How good is the proposed LEQT model by utilizing different expert and query features? Experiments are conducted to evaluate different versions of the proposed model with different types of features.

3) How does the proposed LEQT model work with different document retrieval methods? Experiments are conducted to evaluate the proposed model when it is provided with different document retrieval methods for single data source retrieval.

5.1 Experimental results compared with results obtained from prior research

The section compares the performance of the proposed discriminative models with that of three prior methods. Table 3 summarizes the results. The “Concatenation” method represents the combination strategy presented in the P@NOPTIC system [11], which essentially treats every information source with equal weights. “expCombSUM” and “expCombMNZ” are two data fusion methods proposed in [23] for expert finding and they have shown good performance among the 11 voting schemes[4]. The other four methods in the table are the discriminative models proposed in this paper.

Table 3: Comparison of the experimental results of the proposed discriminative models with the results obtained from prior research. The †symbol indicates statistical significance at 0.9 confidence interval

	$P@5$	$P@10$	$P@15$	$P@20$
Concatenation	0.653	0.592	0.548	0.522
expCombSUM	0.684	0.626	0.608	0.562
expCombMNZ	0.665	0.621	0.596	0.549
EQInd	0.723	0.654	0.630	0.604
LEC	0.771	0.690	0.651	0.646
LEQT	0.816	0.737	0.664	0.650

We can see from Table 3 that “expCombSUM” and “expCombMNZ” can improve upon “Concatenation”. Between them, the performance of “expCombSUM” is slightly better than that of “expCombMNZ”. With the aid of the training set, “EQInd” that uses learned wights is superior to “expCombSUM” and “expCombMNZ”. Furthermore, by introducing the expert features and allowing the combination weights to vary across different experts, additional improvements are achieved by the proposed expert dependent model. Similarly, by introducing the query features alone also improves upon EQInd. Finally, by having both expert and query dependencies, we can achieve the best performance in all the four cases.

5.2 Experimental results by utilizing different types of features

In this experiment, the expert and query dependent model is tested on different sets of features. As shown in Table 2, the total features are divided into four sets. We remove the first three sets of features from the whole respectively and experiment on the resulting features accordingly. Table 4 includes the comparisons against the model with all the features (All). It is not surprising to see that the utilization of all the features yields the best result. The performance does not deteriorate too much after removing the category features (F3) from the full feature set, which indicates that the F3 features are weak. On the other hand, the expert and query statistics feature set (F2) seem more indicative. In addition, the source indicators (F1) seem quite discriminative given that the total number of them is 4, which is relatively small. By comparing Table 4 with Table 3, we can find that LEQT performed always better than EQInd no matter which feature set is used in LEQT. This observation suggests that the expert and query independent model has limited effectiveness by keeping combination strategy constant for different expert and query topics.

5.3 Experimental results by utilizing different document retrieval methods

In this experiment, we use three different document retrieval models to assess the extent to which the performance of the proposed discriminative model is affected by the choice of the underlying document retrieval model. Table 5 shows the retrieval performance of the proposed expert and query probabilistic model across three retrieval models, which are

[4]In this experiment, instead of aggregating scores over documents as in [23], we use “expCombSUM” and “expCombMNZ” to aggregate scores over data sources

Table 4: Experimental results of the LEQT model by utilizing different types of features. "All-X" denotes the remaining features after removing the feature set X from all the features

	P@5	P@10	P@15	P@20
All-F1	0.742	0.672	0.645	0.621
All-F2	0.728	0.664	0.636	0.615
All-F3	0.770	0.701	0.654	0.639
All	0.816	0.737	0.664	0.650

BM25 [29], PL2 [27], and the default Indri retrieval model (i.e., Indri language modeling and inference networks [32]). The full set of features is used in the experiment. From the table, we can see that the performances on the different retrieval models are quite similar, which indicates that the LEQT model is robust to the underlying document retrieval model. On the other hand, by comparing Table 5 with Table 3, we can observe that LEQT with different retrieval models always yielded better performance than EQInd and LEC with the default Indri retrieval model. This observation suggests that the improvements of LEQT over EQInd and LEC do not come from the underlying retrieval model, but from the capture of the latent expert classes and query topics.

Table 5: Experimental results of the LEQT model by utilizing different document retrieval methods

	P@5	P@10	P@15	P@20
BM25	0.820	0.738	0.651	0.644
PL2	0.824	0.745	0.650	0.638
Indri	0.816	0.737	0.664	0.650

6. CONCLUSIONS AND FUTURE RESEARCH

Expert finding is an interesting research problem with many important applications. In this task, the evidence for expertise usually comes from heterogeneous knowledge sources, which poses a key challenge to the task. Although many learning to rank methods have been developed and successfully applied to ad-hoc retrieval, none of them has been explicitly proposed for expert finding. In this paper, we propose a discriminative learning framework along with three probabilistic models by treating expert finding as a knowledge source combination problem. The framework is essentially based on the logistic regression model to estimate the conditional probability of relevance. The proposed LEQT model is capable to adapt the combination strategy to specific queries and experts, which leads to much flexibility of combining data sources when dealing with a broad range of expertise areas and a large variation in experts. The parameter estimation can be efficiently done in EM algorithms. An extensive set of experiments have been conducted on the INDURE testbed to show the effectiveness and robustness of the proposed probabilistic models.

There are several directions to improve the research in this work. First of all, there exists some useful query or expert information that cannot be described by explicit feature representation such as the vector representation used in the paper. For example, the similarity between queries can be sometimes calculated from the taxonomy of expertise or the similarity between experts can be derived based on expert affiliations. By utilizing the well-known "kernel trick", instead of designing explicit features, we can transform the similarity metric between queries or experts into the implicit feature space in the form of a Mercer kernel. The kernel versions of the proposed models can then be very desirable. Secondly, the proposed discriminative learning models can also serve as the building block for other important IR problems such as query expansion and active learning in the context of expert finding. The applicability of the LEQT model is even not limited to the expert finding problems. It can also be used in many other areas involving knowledge source combination, such as distributed information retrieval, question answering, cross-lingual information retrieval, and multi-sensor fusion.

7. REFERENCES

[1] J. Aslam and M. Montague. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 276–284. ACM New York, NY, USA, 2001.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–50. ACM New York, NY, USA, 2006.

[3] K. Balog, T. Bogers, L. Azzopardi, M. de Rijke, and A. van den Bosch. Broad expertise retrieval in sparse data environments. In *Proceedings of the 30th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 551–558. ACM New York, NY, USA, 2007.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 89–96, 2005.

[5] J. Callan, Z. Lu, and W. Croft. Searching distributed collections with inference networks. In *Proceedings of the 18th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–28. ACM New York, NY, USA, 1995.

[6] C. Campbell, P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In *Proceedings of the 12th International Conference on Information and Knowledge Management*, pages 528–531. ACM New York, NY, USA, 2003.

[7] Z. Cao, T. Qin, T. Liu, M. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, pages 129–136. ACM New York, NY, USA, 2007.

[8] K. Crammer and Y. Singer. Pranking with ranking. *Advances in Neural Information Processing Systems*, 1:641–648, 2002.

[9] N. Craswell. W3C Test Collection, 2005.

[10] N. Craswell, A. de Vries, and I. Soboroff. Overview of

the trec-2005 enterprise track. In *TREC 2005 Conference*, pages 199–205, 2005.
[11] N. Craswell, D. Hawking, A. Vercoustre, and P. Wilkins. P@ noptic expert: Searching for experts not just for documents. In *Ausweb Poster Proceedings, Queensland, Australia*, 2001.
[12] T. Davenport and L. Prusak. Working knowledge: How organizations manage what they know. *Ubiquity*, 1(24), 2000.
[13] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.
[14] H. Fang and C. Zhai. Probabilistic models for expert finding. In *Proceedings of the 29th European Conference on Information Retrieval*, pages 418–430, 2007.
[15] E. Fox and J. Shaw. Combination of multiple searches. In *Proceedings of the 2nd Text REtrieval Conference (TREC)*, pages 243–243. National Institute of Standards and Technology, 1994.
[16] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4:933–969, 2003.
[17] J. Gao, H. Qi, X. Xia, and J. Nie. Linear discriminant model for information retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 290–297. ACM New York, NY, USA, 2005.
[18] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142. ACM, 2002.
[19] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM Conference on Knowledge Discovery and Data Mining*, 2002.
[20] I. Kang and G. Kim. Query type classification for web document retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71. ACM New York, NY, USA, 2003.
[21] L. Kennedy, A. Natsev, and S. Chang. Automatic discovery of query-class-dependent models for multimodal search. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 882–891. ACM New York, NY, USA, 2005.
[22] T. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, 2007.
[23] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM CIKM*, pages 387–396. ACM New York, NY, USA, 2006.
[24] A. Mockus and J. Herbsleb. Expertise browser: a quantitative approach to identifying expertise. In *Proceedings of the 24th International Conference on Software Engineering*, pages 503–512. ACM New York, NY, USA, 2002.
[25] R. Nallapati. Discriminative models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71. ACM New York, NY, USA, 2004.
[26] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 2:841–848, 2002.
[27] V. Plachouras, B. He, and I. Ounis. University of Glasgow at TREC2004: Experiments in web, robust and terabyte tracks with Terrier. In *Proceedings of the 13th Text REtrieval Conference (TREC)*, 2005.
[28] T. Qin, X. Zhang, M. Tsai, D. Wang, T. Liu, and H. Li. Query-level loss functions for information retrieval. *Information Processing and Management*, 44(2):838–855, 2008.
[29] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-4. In *Proceedings of the 4th Text Retrieval Conference (TREC)*, pages 73–97, 1996.
[30] J. Savoy, A. Le Calvé, and D. Vrajitoru. Report on the TREC-5 Experiment: Data Fusion and Collection Fusion. In *Proceedings of the 5th Text REtrieval Conference (TREC)*, pages 489–502. National Institute of Standards and Technology, 1997.
[31] P. Serdyukov and D. Hiemstra. Modeling documents as mixtures of persons for expert finding. In *Proceedings of 30th European Conference on Information Retrieval*, volume 4956, page 309. Springer, 2008.
[32] T. Strohman, D. Metzler, H. Turtle, and W. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*, 2004.
[33] C. Vogt and G. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.
[34] C. Vogt, G. Cottrell, R. Belew, and B. Bartell. Using relevance to train a linear mixture of experts. In *Proceedings of the 5th Text REtrieval Conference (TREC)*, pages 503–515. National Institute of Standards and Technology, 1997.
[35] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 391–398. ACM New York, NY, USA, 2007.
[36] R. Yan and A. Hauptmann. Probabilistic latent query analysis for combining multiple retrieval sources. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 324–331. ACM New York, NY, USA, 2006.
[37] R. Yan, J. Yang, and A. Hauptmann. Learning query-class dependent weights in automatic video retrieval. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, pages 548–555. ACM New York, NY, USA, 2004.