Inf Retrieval (2011) 14:158–177
DOI 10.1007/s10791-010-9139-3

Discriminative probabilistic models for expert search in heterogeneous information sources

Yi Fang · Luo Si · Aditya P. Mathur

Received: 24 April 2009 / Accepted: 11 June 2010 / Published online: 21 August 2010

Abstract In many 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 information 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 discriminative 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/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. To the best of our knowledge, this is the first work that designs discriminative learning models to rank experts. Empirical studies on two real world faculty expertise testbeds demonstrate the effectiveness and robustness of the proposed discriminative learning models.

Keywords Expert finding · Expert search · Learning to rank

Y. Fang (✉) · L. Si · A. P. Mathur
Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA
e-mail: fangy@cs.purdue.edu

L. Si
e-mail: lsi@cs.purdue.edu

A. P. Mathur
e-mail: apm@cs.purdue.edu

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 four 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 three 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 some 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. This paper provides thorough experimental results as well as detailed analysis, which extends the preliminary research in (Fang et al. 2009). In the experiments, the proposed discriminative models have shown to have better performance than the prior solutions on two real world faculty expertise testbeds (i.e., the Indiana Database of University Research Expertise (INDURE)[1] (Fang et al. 2008) and the UvT Expert collection (Balog et al. 2007). 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 proposes different discriminative probabilistic models for expert search in heterogeneous information sources. Section 4 presents the experimental results and the corresponding discussions. Section 5 concludes.

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

2 Related work

Initial approaches to expert finding employed a manually constructed database which listed experts by category and subcategory (Davenport and Prusak 2000). 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 (Mockus and Herbsleb,2002) and email (Campbell et al. 2003). With abundant information becoming available on the Web, there is increasing interest in utilizing varied and heterogeneous sources of expertise evidence (Balog et al. 2007).

Expert finding has attracted a lot of interest in the IR community since the launch of Enterprise Track (Craswell et al. 2005) at TREC and rapid progress has been made in modeling and evaluations. Most of the previous work on TREC expert finding task generally fall into two categories: profile-centric and document-centric approaches. Profile-centric approaches build an expert representation by concatenating all the documents or text segments associated with that expert. The user query is matched against this representation and thus finding experts is equal to retrieve documents. The document-centric approaches are instead based on the analysis of individual documents. Balog et al. (2006) formalize the two methods. Their Model 1 directly models the knowledge of an expert from associated documents, which is equivalent to a profile-centric approach, and Model 2 first locates documents on the topic and then finds the associated experts, which is a document-centric approach. Petkova and Croft (2007) has further improved their models by proposing a proximity-based document representation for incorporating sequential information in text. There are many generative probabilistic models proposed for expert finding. For example, Serdyukov and Hiem- stra (2008) propose an expert-centric language model and Fang and Zhai (2007) apply the probabilistic ranking principle to the expert search task. Cao et al. (2005) propose a two-stage language model combining a document relevance and co-occurrence model. The generative probabilistic framework naturally lends itself to many extensions such as including document and candidate evidence through the use of document structure (Zhu et al. 2006) and hierarchical structure (Petkova and Croft 2006). Mac- donald and Ounis (2006) treats the problem of ranking experts as a voting problem and explored 11 different voting strategies to aggregate over the documents associated with the expert. However, previous approaches do not differentiate data sources, which may cause unsatisfactory performance in real world applications where some data sources are likely more reliable and indicative than others.

The collection used in expert finding task in TREC 2005 and 2006 represents the internal documentation of the World Wide Web Consortium (W3C) and was crawled from the public W3C (*.w3.org) sites in June 2004 (Craswell et al. 2005). In the 2007 edition of the TREC Enterprise track, CSIRO Enterprise Research Collection (CERC) (Bailey et al. 2007) was used as the document collection. In these two testbeds, the relationship between documents and experts is ambiguous and therefore a large amount of effort in previous expert finding research is devoted to model the document-expert associations. In contrast, the UvT Expert collection (Balog et al. 2007) is a popular alternative testbed with much broader coverage of expertise areas and clear document-expert associations. The INDURE testbed and UvT testbed share similar characteristics as both of them contain a set of heterogeneous information sources and include certain document-expert relationship. More detailed information about these two testbeds can be found in Sect. 4.

The proposed voting process in expert finding is also closely related to data fusion in metasearch (Aslam and Montague 2001) and collection fusion problem in distributed

information retrieval (Callan et al. 1995). The general retrieval source combination problem has been examined by a significant body of previous work. Fox and Shaw (1994)'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 Savoy et al. (1997); Vogt et al. (1997); Vogt and Cottrell (1999) 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 strategies keep unchanged for different query topics. Recent work (Kang and Kim 2003) 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 (Kennedy et al. 2005; Yan et al. 2004) have been shown superior to query-independent combination. The work that is more closely related to ours is the work done by Yan and Hauptmann (2006). 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 (Liu et al. 2007) 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 (Herbrich et al. 2002; Crammer and Singer 2002). 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 (Freund et al. 2003; Joachims 2002; Burges et al. 2005; Gao et al. 2005; Xu and Li 2007). More recently, the listwise approach, *ListNet* (Cao et al. 2007), 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 (Qin et al. 2008). Although valuable work has been done for learning to rank for ad-hoc retrieval, no research has been conducted for designing discriminative 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. The UvT Expert collection also comes from four data sources (i.e., research descriptions,

course descriptions, publications, and academic homepages). For the document collection, there are totally M experts and the document-expert association is clear (e.g., the supervisors of PhD dissertations, the owners of homepages and the principal investigators of NSF projects). Within a single document source, each expert has a set of 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 ith 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 top supporting documents 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 subset of supporting documents for e in the ith source, and more details are discussed in Sect. 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 model casts expert finding into a binary classification problem that treats the relevant query-expert pairs as positive instances and irrelevant pairs as negative instances. 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 (Ng and Jordan 2002) 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 (Nallapati 2004). 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 $P(r|e, q)$. Our retrieval problem is a two-class classification in the sense that $r \in \{1, -1\}$ in which $r = 1$ indicates the expert e is relevant to the query q and $r = 0$ indicates not relevant. The parametric form of $P(r = 1|e, q)$ can be expressed as follows in terms of logistic functions over a linear function of features

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

where $\sigma(x) = 1/(1 + \exp(-x))$ is the standard logistic function. Here the features are the retrieval scores from individual data sources. ω_i is the combination parameter for the ith data source. For the non-relevance class, we can get

$$P(r = -1|e, q) = 1 - P(r = 1|e, q) = \sigma\left(-\sum_{i=1}^{K} \omega_i s_i(e, q)\right) \tag{2}$$

We can see that for different values of r, the only difference in computing $P(r|e, q)$ is the sign inside the logistic function. In the following sections, we adopt the general representation of $P(r|e, q) = \sigma(r \sum_{i=1}^{K} \omega_i s_i(e, q))$. The experts are then ranked according to the descending order of $P(r = 1|e, q)$. Because the learned weights are identical for all

experts and queries and thus it is also called expert and query independent (EQInd) model in the subsequent sections. This model is also equivalent to logistic regression.

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 their 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. Moreover, 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 to indicate which expert class the combination weights $\omega_{z\cdot} = (\omega_{z1},\ldots, \omega_{zK})$ 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

$$P(r, z|q, e; \alpha, \omega) = P(z|e; \alpha)P(r|q, e, z; \omega) \tag{3}$$

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

$$P(r|q, e; \alpha, \omega) = \sum_{z=1}^{N_z} P(z|e; \alpha)\sigma\left(r\sum_{i=1}^{K}\omega_{zi}s_i(e, q)\right) \tag{4}$$

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 α_{zj} is the weight parameter associated with the jth expert feature in the latent expert class z and 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. (4), we can get

$$P(r|q,e;\alpha,\omega) = \frac{1}{Z_e}\sum_{z=1}^{N_z}\exp\left(\sum_{j=1}^{L_z}\alpha_{zj}e_j\right)\sigma\left(r\sum_{i=1}^{K}\omega_{zi}s_i(e,q)\right) \tag{5}$$

Because α_{zj} is associated with each expert feature instead of each training expert, the above model allows the estimated α_{zj} to be applied to any unseen expert.

3.3.1 Parameter estimation

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

$$l(\omega,\alpha) = \sum_{u=1}^{N}\sum_{v=1}^{M}\log\left(\sum_{z=1}^{N_z}\left(\frac{1}{Z_{e_v}}\exp\left(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj}\right)\right)\sigma\left(r_{uv}\sum_{i=1}^{K}\omega_{zi}s_i(e_v,q_u)\right)\right) \tag{6}$$

where N is the number of queries and M is the number of experts, e_{vj} denotes the jth feature for the vth expert e_v and r_{uv} denotes the relevance judgment for the pair (q_u, e_v). A typical approach to maximizing Eqn. (6) is to use the Expectation-Maximization (EM) algorithm (Dempster et al. 1977), 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,

$$P(z|e_v,q_u) = \frac{\exp\left(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj}\right)\sigma\left(r_{uv}\sum_{i=1}^{K}\omega_{zi}s_i(e_v,q_u)\right)}{\sum_z \exp\left(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj}\right)\sigma\left(r_{uv}\sum_{i=1}^{K}\omega_{zi}s_i(e_v,q_u)\right)} \tag{7}$$

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

$$\omega_{z\cdot}^{*} = \arg\max_{\omega_{z\cdot}}\sum_{uv}P(z|e_v,q_u)\log\left(\sigma\left(\sum_{i=1}^{K}\omega_{zi}s_i(e_v,q_u)\right)\right) \tag{8}$$

$$\alpha_{z\cdot}^{*} = \arg\max_{\alpha_{z\cdot}}\sum_{u}\left(\sum_{v}P(z|e_v,q_u)\right)\log\left(\frac{1}{Z_{e_v}}\exp\left(\sum_{j=1}^{L_z}\alpha_{zj}e_{vj}\right)\right) \tag{9}$$

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. The number of expert classes can be obtained by maximizing the sum of log-likelihood and some model selection criteria. In our work, we choose Akaike Information Criteria (AIC) (Akaike 1974) as the selection criterion, which has been shown to be suitable in determining the number of latent classes in mixture

models (McLachlan and Peel 2004). It is a measure of the goodness of fit of an estimated statistical model, which is defined in the general case as follows

$$2l(\omega, \alpha) - 2m \tag{10}$$

where m is the number of parameters in the statistical model. The second term in AIC corresponds to a model-complexity regularization, which has a solid ground in information theory. 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 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 the NSF projects 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. Similar to the LEC model, we propose the latent query topic (LQT) model by using a latent variable t to indicate the topic that the query comes from. Thus, the weight ω_{ti} now depends on query t.

The mixing proportions $P(t|q;\beta)$ can also be modeled using $\frac{1}{T_q}\exp(\sum_{g=1}^{L_t}\beta_{tg}q_g)$ where L_t is the number of query features, q_g is the gth query feature for query q, β_{tg} is the weight parameter associated with the gth query feature in the latent query topic t, T_q is the normalization factor that scales the exponential function to be a probability distribution. The corresponding mixture model can be written as

$$P(r|q,e;\alpha,\omega) = \frac{1}{T_q}\sum_{t=1}^{N_t}\exp\left(\sum_{g=1}^{L_t}\beta_{tg}q_g\right)\sigma\left(r\sum_{i=1}^{K}\omega_{ti}s_i(e,q)\right) \tag{11}$$

where N_t is the number of latent query topics and ω_{ti} is the weight for the ith information source s_i under the topic t. The parameters can be estimated similarly by EM algorithm as in LEC.

3.5 Expert and query dependent probabilistic models

Based on the dependence of the combination strategy on both experts and queries, it is natural to combine LEC and LQT into a single probabilistic model, which we call 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,

$$P(r,z,t|q,e) = P(t|q)P(z|e)P(r|q,e,z,t) \tag{12}$$

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

$$P(r|q,e) = \sum_{t=1}^{N_t} \sum_{z=1}^{N_z} P(t|q)P(z|e)\sigma\left(r \sum_{i=1}^{K} \omega_{zti} s_i(e,q)\right) \tag{13}$$

where ω_{zti} is the weight for s_i under the expert class z and query topic t. By plugging the soft-max functions into $P(z|e;\alpha)$ and $P(t|q;\beta)$, Eqn. (13) can then be reformulated as

$$P(r|q,e) = \frac{1}{Z_e T_q} \sum_{t=1}^{N_t} \sum_{z=1}^{N_z} \exp\left(\sum_{j=1}^{L_z} \alpha_{zj} e_j\right) \exp\left(\sum_{g=1}^{L_t} \beta_{tg} q_g\right) \sigma\left(r \sum_{i=1}^{K} \omega_{zti} s_i(e,q)\right) \tag{14}$$

The LEQT model combines the advantages of both LEC and LQT. When $N_t = 1$, LEQT degenerates to LEC and similarly when $N_z = 1$, it degrades to LQT. When both numbers are equal to 1, LEQT becomes the logistic regression model in Sect. 3.2. Therefore, LEC, LQT 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,

$$P(z,t|e_v,q_u) = \frac{\exp\left(\sum_{\mathrm{j}=1}^{\mathrm{L_z}} \alpha_{\mathrm{zj}} \mathrm{e_{vj}}\right) \exp\left(\sum_{\mathrm{g}=1}^{\mathrm{L_t}} \beta_{\mathrm{tg}} \mathrm{q_{ug}}\right) \sigma\left(\mathrm{r_{uv}} \sum_{\mathrm{i}=1}^{\mathrm{K}} \omega_{\mathrm{zti}} \mathrm{s_i(e_v, q_u)}\right)}{\sum_{zt} \exp\left(\sum_{j=1}^{L_z} \alpha_{zj} e_{vj}\right) \exp\left(\sum_{g=1}^{L_t} \beta_{tg} q_{ug}\right) \sigma\left(r_{uv} \sum_{i=1}^{K} \omega_{zti} s_i(e_v, q_u)\right)} \tag{15}$$

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

$$\omega_{zt\cdot}^{*} = \arg\max_{\omega_{zt\cdot}} \sum_{uv} P(z,t|e_v,q_u) \log\left(\sigma\left(\sum_{i=1}^{K} \omega_{zti} s_i(e_v,q_u)\right)\right) \tag{16}$$

$$\alpha_{z\cdot}^{*} = \arg\max_{\alpha_{z\cdot}} \sum_{v} \left(\sum_{ut} P(z,t|e_v,q_u)\right) \log\left(\frac{1}{Z_{e_v}} \exp\left(\sum_{j=1}^{L_z} \alpha_{zj} e_{vj}\right)\right) \tag{17}$$

$$\beta_{t\cdot}^{*} = \arg\max_{\beta_{t\cdot}} \sum_{u} \left(\sum_{vz} P(z,t|e_v,q_u)\right) \log\left(\frac{1}{T_{q_u}} \exp\left(\sum_{g=1}^{L_t} \beta_{tg} q_{ug}\right)\right) \tag{18}$$

3.6 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

Table 1 Four types of features used in the experiments by the proposed models

Source indicator	Whether each data source is absent for the given expert
Expert and query statistics	Number of supporting documents for the expert within each data source (e.g., number of publications, number of NSF proje;cts, and number of supervised PhD dissertations associated with the expert)
	Given a query, the number of documents retrieved for each data source
	Given a query, the mean and variance of the number of supporting documents for retrieved experts within each data source
	The normalized length (in the number of words) of supporting documents within each data source for the given expert
	Variance of the above numbers
	Number of words in the query
Category	Posterior probabilities of the expert and query belonging to the eight predefined classes
Others	Number of outgoing links in the homepage
	Whether faculty homepage contains certain keywords such as "distinguished professor", "assistant professor", etc
	Number of images in the homepage

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). Table 1 contains more details of the features that we used in the experiments.

4 Experiments

In the experiments, we evaluate the effectiveness of the proposed models on the INDURE and UvT testbeds. These two data collections share similar characteristics, but differ from the TREC data sets for expert finding (i.e., W3C and CSIRO). In INDURE and UvT, the data come from multiple information sources and document-author associations are clear. In addition, both collections cover a broad range of expertise areas.

We apply the Indri retrieval model (Strohman et al. 2004) as the default document retrieval method to obtain the single source retrieval score $s_i(d, q)$. The Indri toolbox[2] is used in the experiments. The total features can be divided into four sets as presented in Table 1: (1) source indicators that show whether each data source is 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 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. Both INDURE and UvT collections use roughly the same set of features with minor difference as some features for INDURE are not applicable for UvT (e.g., the number of NSF projects) and vice versa. As a result, there are 21 query features and 34 expert features for the INDURE collection, and 20 query features and 32

[2] http://www.lemurproject.org/indri/.

expert features for UvT. 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.

An extensive set of experiments were designed on the two testbeds 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.

4.1 Retrieval evaluation for the INDURE faculty expertise collection

The INDURE faculty expertise collection used in the experiments is constructed from the 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 (PR); (2) faculty homepages (HP); (3) NSF funding project descriptions (NSF); (4) faculty publications and supervised PhD dissertations (PUB). The profiles include faculty research areas, which could be keywords from a predefined taxonomy[3] or free keywords that adequately describe the expertise.

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. Mathematically, $s_i(e,q) = \sum_{d \in top(e,k)} s_i(d,q)$, where $top(e, k)$ denotes the set of top-k scored documents for e. In the experiments, we choose $k = 20$. To train and test the proposed models, 50 training queries and 50 testing queries were selected from the query log and Table 2 includes a subset of them. For each training query, we examine the list of results returned from the "Concatenation" ranking method (discussed in Sect. 4.1.1) and judge at most 80 experts as the positive instances and as the negative ones respectively. 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 3 contains some statistics of the testbed.

As discussed in Sect. 3.3, the numbers of latent variables in the proposed models are set by optimizing the AIC criteria. Because the training data are limited, a large number of

[3] http://www.indure.org/hierarchy.cfm.

Table 2 A subset of queries with relevance judgments used for evaluation

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

Table 3 Descriptive statistics of the INDURE faculty expertise collection

Total number of experts	12,535
Number of training queries	50
Number of testing queries	50
Number of training experts	3,154
Total number of expert-query relevance judgments	6,482
Average number of training experts per query	130
Maximum number of training experts per query	160
Minimum number of training experts per query	52
Average number of queries per expert	2.1
Number of training experts with PR	3,154
Number of training experts with HP	1,251
Number of training experts with NSF	306
Number of training experts with PUB	1,842

Table 4 Number of latent variables determined by AIC for INDURE

	N_z	N_t
LEC	9	N/A
LQT	N/A	6
LEQT	6	5

parameters may cause the proposed probabilistic models to overfit. Therefore, in the experiments, we maximize AIC with respect to N_z and N_t in the range from 1 to 10. Table 4 presents the numbers of latent variables chosen for INDURE.

4.1.1 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 5 summarizes the results. The "Concatenation" method represents the combination strategy presented in the P@NOPTIC system (Craswell et al. 2001), which essentially treats every information source with equal weights. "expCombSUM" and "expCombMNZ" are two data fusion methods proposed in (Macdonald and Ounis 2006) for expert finding and they have shown good performance among the 11 voting schemes. The other four methods in the table are the discriminative models proposed in this paper.

The "Model 2" method in the table refers to the retrieval model originally proposed in (Balog et al. 2006) and it is one of the most effective formal methods for expert search. We can see from Table 5 that all the proposed models outperform Model 2. Moverover, "expCombSUM" and "expCombMNZ" can improve upon "Concatenation". Between

Table 5 Comparison of the experimental results of the proposed discriminative models with the results obtained from prior research

	P@5	*P*@10	*P*@15	*P*@20
Model 2	0.696	0.633	0.604	0.571
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†
LQT	0.762	0.678†	0.648†	0.638†
LEQT	0.816†	0.737†	0.664†	0.650†

† Statistical significance at 0.9 confidence interval

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. In this case study, LEC generally performed better than LQT, but their difference is not substantial. Finally, by having both expert and query dependencies, we can achieve the best performance in all the four cases. To provide more detailed information, we do statistical significance testing between "Concatenation" and other methods by the sign s-tests and results are also reported in the table.

In addition, we examined some cases in which the ranking is improved by LEQT and found the intuitions of the proposed latent variable models are manifested in these cases. For example, Prof. Melvin Leok is not ranked highly by the "Concatenation" and "EQInd" methods for the query "numerical analysis", although he is a well-known young researcher in this area. We found that part of the reason is he does not have supervised PhD dissertation data, which causes his final merged retrieval score less comparable with those who have all sorts of information. On the other hand, the LEQT model can rank him in top part of the list by shifting the weights from PhD dissertations to his multiple NSF projects and homepage. We also observed that some other cases are also helped by the proposed models such as those stated in previous sections as the motivations of the work. However, we do find that this shift-of-weight scheme can sometimes cause undesirable effect. For example, some faculty do not have NSF projects not because the projects are not applicable for them, but maybe because they are not competent enough to get funded by NSF yet. In this case, the shift of weight may exaggerate the importance of other data sources and hurt the retrieval performance.

4.1.2 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 1, 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 6 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

Table 6 Experimental results of the LEQT model by utilizing different types of 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

"All-X" denotes the remaining features after removing the feature set X from all the features

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 6 with Table 5, 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.

4.1.3 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 7 shows the retrieval performance of the proposed expert and query probabilistic model across three retrieval models, which are BM25 (Robertson et al. 1996), PL2 (Plachouras et al. 2005), and the default Indri retrieval model (i.e., Indri language modeling and inference networks (Strohman et al. 2004)). The full set of features is used in the experiment. From the table, we can see that the performance 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 7 with Table 5, we can observe that LEQT with different retrieval models always yielded better performance than EQInd, LQT and LEC with the default Indri retrieval model. This observation suggests that the improvements of LEQT over EQInd, LQT and LEC do not come from the underlying retrieval model, but from capturing the latent expert classes and query topics.

4.2 Retrieval evaluation for the UvT expert collection

In this section, we experiment on the existing UvT Expert collection which has been developed for expert finding and expert profiling tasks. The collection is based on the Webwijs ("Webwise") system developed at Tilburg University (UvT) in the Netherlands.

Table 7 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

Similar to INDURE, there are four data sources in UvT: research descriptions (RD), course descriptions (CD), publications (PUB), and academic homepages (HP). Webwijs is available in Dutch and English. Not all Dutch topics/queries have an English translation, but every Dutch page has an English translation. In our experiments, we only use the English data for evaluation.

To train our proposed model, we randomly select 200 topics as the training queries among the total 981 topics. Because the expertise topics in UvT are self-selected by experts, we can get the relevant experts for each selected topic, which are viewed as the positive instances for our discriminative training. To obtain a set of negative instances, we use the "Concatenation" method introduced in Sect. 4.1.1 to retrieve a list of candidate experts for each selected query. Excluding the positive experts from the list, we choose the same number of the top ranked experts as negative experts for the query. We test the proposed models on the rest 781 topics and corresponding relevant experts. The evaluation measures are Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). Table 8 contains the statistics of the data we used in our experiments. We follow the similar procedure with that in INDURE to set the number of latent variables in the proposed models. Table 9 presents the numbers of latent variables chosen for UvT.

4.2.1 Experimental results compared with results obtained from prior research

This section compares the performance of the proposed discriminative models with that of prior methods. Table 10 summarizes the results. The columns of the table correspond to the combinations of various data sources (RD, CD, PUB, and HP) and RD+CD+PUB+HP is equivalent to the full collection. The "Model 2" was evaluated on the UvT Expert data

Table 8 Descriptive statistics of the UvT expert collection

	All	Training
Number of experts	1,168	328
Number of topics	981	200
Number of expert-topic pairs	3,251	685
Total number of expert-topic relevance judgement	N/A	1,359
Number of experts with at least one topic	743	328
Average number of topics/expert	5.9	2.1
Maximum number of topics/expert	35	35
Minimum number of topics/expert	1	1
Average number of experts/topic	3.3	3.43
Maximum number of experts/topic	30	16
Minimum number of experts/topic	1	1
Number of experts with HP	318	98
Number of experts with CD	318	86
Number of experts with RD	313	95
Number of experts with PUB	734	209
Average number of PUBs per expert	27.0	28.3
Average number of PUB citations per expert	25.2	26.2
Average number of full-text PUBs per expert	1.8	1.9

Table 9 Number of latent variables determined by AIC for UvT

	N_z	N_t
LEC	7	N/A
LQT	N/A	5
LEQT	5	3

Table 10 Comparison of the experimental results of the proposed discriminative models with the results obtained from prior research

	RD+CD		RD+CD+PUB		RD+CD+PUB+HP	
	MAP	MRR	MAP	MRR	MAP	MRR
Model 2	0.201	0.365	0.271	0.432	0.286	0.446
Concatenation	0.193	0.358	0.262	0.421	0.274	0.425
expCombSUM	0.198	0.355	0.264	0.425	0.280	0.431
expCombMNZ	0.195	0.351	0.269	0.428	0.286	0.429
EQInd	0.221	0.372	0.301	0.457	0.325†	0.469†
LEC	0.242†	0.389†	0.332†	0.472†	0.362†	0.486†
LQT	0.234	0.366	0.315†	0.467†	0.341†	0.477†
LEQT	0.254†	0.397†	0.343†	0.476†	0.371†	0.498†

The columns correspond to the combinations of various data sources

† Statistical significance at 0.9 confidence interval

collection and achieved relatively better performance than the other methods as reported in (Balog et al. 2007).

As we can see from the table, the results roughly follow the same pattern with the previous evaluation on INDURE as shown in Table 5. The learning approaches improve the performance over those which do not differentiate information sources and the latent variables can bring additional gains by shifting the weights according to specific experts and queries. In particular, all the proposed models outperform Model 2 which shows good performance on the other expert search testbeds. Model 2 performs slightly better than the heuristic combination methods (i.e., "Concatenation", "expCombSUM" and "expCombMNZ"), but their differences are not significant. On the other hand, as more heterogeneous data sources are incorporated, the improvement brought by proposed models over the baseline seem more significant.

To examine the specific queries that have improved performance, we find that the flexible combination strategies do help. For example, the topic "literature (1585)" has many occurrences in the course descriptions which are no indication of expertise in this area (e.g., the required literature finding/review for the course). The Model 2 and EQInd methods yields low AP and RR performance, because some irrelevant experts with these course descriptions are retrieved among the top. In contrast, the LEQT method boosts the rank of the relevant experts by downweighting the course description for this query. Similar to the INDURE evaluation, the shift-of-weight effect is also observed on many experts who have missing sources. For example, for the topic "machine learning", the expert 986356 is relevant, but is not ranked at the top by either Model 2 or the EQInd method. The reason is that the expert has no course description and homepage available in the collection, although he has intensive publications on this topic. On the other hand, the

top ranked expert has complete information from all the four sources although he is actually not relevant to this topic. LEQT reverses the ranks of these two experts and consequently improves AP and RR for this query.

4.2.2 *Experimental results by utilizing different types of features and different document retrieval methods*

The LEQT model is tested on different sets of features in the same way to the INDURE evaluation as shown in Sect. 4.1.2. Table 11 includes the results. They generally follow the similar pattern as those in Table 6, but we can find that the F1 features become stronger discriminators. This may come from the fact that the data source missing problem is more pervasive in UvT than in INDURE as we can see from Table 8 that there exist a significant number of people who do not have data for each data source. This makes the shift-of-weight effect on missing sources more desirable.

Similar to Table 7, we show how robust of the proposed models with respect to the choice of the underlying document retrieval model and Table 12 contains the corresponding results, which are consistent with the results presented in Table 7. The results suggest that the gains in performance are not from the specific document retrieval methods, but from the flexible combination strategy of the proposed probabilistic models.

5 Conclusions and future research

Expert finding in an organization is an important task and discovering the relevant experts given a topic can be very challenging, particularly in many realistic settings where the evidence for expertise comes from heterogeneous knowledge sources. Although many learning to rank methods have been developed and successfully applied to ad-hoc retrieval, none of them has been proposed for expert finding. In this paper, we propose a discriminative learning framework along with four probabilistic models by treating expert finding as a knowledge source combination problem. 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 and

Table 11 Experimental results of the LEQT model by utilizing different types of features

	All-F1	All-F2	All-F3	All
MAP	0.346	0.334	0.366	0.371
MRR	0.479	0.473	0.491	0.498

"All-X" denotes the remaining features after removing the feature set X from all the features

Table 12 Experimental results of the LEQT model by utilizing different document retrieval methods

	BM25	PL2	Indri
MAP	0.352	0.344	0.371
MRR	0.487	0.465	0.498

UvT testbeds 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, we can refine the proposed models by exploiting knowledge area similarity and contextual information, as the advanced models with these two features have been shown to bring significant improvements over the baseline on the UvT collection (Balog et al. 2007). In certain scenarios, the expert social network can be readily obtained such as co-authors of publications, which is also potentially useful for expert finding. Moreover, it is worthwhile exploring state-of-the-art learning to rank algorithms for expert search, as many of them have demonstrated effectiveness for ad-hoc retrieval. For example, it can be a natural extension to encode the latent expert and query topics into Ranking SVM (Herbrich et al. 2002). Furthermore, it is interesting to go beyond classification models by exploring pairwise or listwise approaches as the training instances of document pairs can be easily obtained in some scenarios. In addition, 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 problem. 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.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*(6), 716–723.

Aslam, J., & Montague, M. (2001). Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 276–284). NY, USA: ACM New York.

Bailey, P., Craswell, N., Soboroff, I., & de Vries, A. (2007). The CSIRO enterprise search test collection. In *ACM SIGIR forum* (pp. 42–45).

Balog, K., Azzopardi, L., & de Rijke, M. (2006). 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* (pp. 43–50). NY, USA:ACM New York.

Balog, K., Bogers, T., Azzopardi, L., & de Rijke, M., van den Bosch, A. (2007). Broad expertise retrieval in sparse data environments. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 551–558). NY, USA: ACM New York.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on machine learning* (pp. 89–96).

Callan, J., Lu, Z., & Croft, W. (1995). Searching distributed collections with inference networks. In *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 21–28). NY, USA: ACM New York.

Campbell, C., Maglio, P., Cozzi, A., & Dom, B. (2003) Expertise identification using email communications. In *Proceedings of the 12th international conference on information and knowledge management* (pp. 528–531). NY, USA: ACM New York.

Cao, Y., Liu, J., Bao, S., & Li, H. (2005). Research on expert search at enterprise track of TREC 2005. In *Proceedings of 14th text retrieval conference (TREC 2005)*.

Cao, Z., Qin, T., Liu, T., Tsai, M., & Li, H. (2007). Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th international conference on machine learning* (pp. 129–136). ACM New York, NY, USA.

Crammer, K., & Singer, Y. (2002). Pranking with ranking. *Advances in Neural Information Processing Systems, 1*, 641–648.

Craswell, N., Hawking, D., Vercoustre, A., & Wilkins, P. (2001). P@ noptic expert: Searching for experts not just for documents. In *Ausweb poster proceedings*. Queensland, Australia.

Craswell, N., de Vries, A., & Soboroff, I. (2005). Overview of the trec-2005 enterprise track. In *TREC 2005 conference* (pp. 199–205).

Davenport, T., & Prusak, L. (2000). Working knowledge: How organizations manage what they know. *Ubiquity, 1*(24).

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*, 1–38.

Fang, H., & Zhai, C. (2007). Probabilistic models for expert finding. In *Proceedings of the 29th European conference on information retrieval* (pp. 418–430).

Fang, Y., Si, L., & Mathur, A. (2008). FacFinder: Search for expertise in academic institutions. Technical Report: SERC-TR-294, Department of Computer Science, Purdue University.

Fang, Y., Si, L., & Mathur, A. (2009). Ranking experts with discriminative probabilistic models. In *Proceedings of SIGIR 2009 workshop on learning to rank for information retrieval*.

Fox, E., & Shaw, J. (1994). Combination of multiple searches. In *Proceedings of the 2nd text retrieval conference (TREC)* (pp. 243–243). National Institute of Standards and Technology.

Freund, Y., Iyer, R., Schapire, R., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *The Journal of machine learning research, 4*, 933–969.

Gao, J., Qi, H., Xia, X., & Nie, J. (2005). Linear discriminant model for information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 290–297). NY, USA: ACM New York.

Herbrich, R., Graepel, T., & Obermayer, K. (2002). Large margin rank boundaries for ordinal regression. In *Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining, ACM* (pp. 133–142).

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM conference on knowledge discovery and data mining*.

Kang, I., & Kim, G. (2003). Query type classification for web document retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 64–71), NY, USA: ACM New York.

Kennedy, L., Natsev, A., & Chang, S. (2005). Automatic discovery of query-class-dependent models for multimodal search. In *Proceedings of the 13th annual ACM international conference on multimedia* (pp. 882–891). NY, USA: ACM New York.

Liu, T., Xu, J., Qin, T., Xiong, W., & Li, H. (2007). 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.*

Macdonald, C., & Ounis, I. (2006). Voting for candidates: Adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM international conference on information and knowledge management* (pp. 387–396). NY, USA: ACM New York.

McLachlan, G., & Peel, D. (2004). Finite mixture models. Chichester: Wiley.

Mockus, A., & Herbsleb, J. (2002). Expertise browser: A quantitative approach to identifying expertise. In *Proceedings of the 24th international conference on software engineering* (pp. 503–512). NY, USA: ACM New York.

Nallapati, R. (2004). Discriminative models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 64–71). NY, USA: ACM New York.

Ng, A., & Jordan, M. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems, 2*, 841–848.

Petkova, D., & Croft, W. (2006). Hierarchical language models for expert finding in enterprise corpora. In *18th IEEE international conference on tools with artificial intelligence, 2006. ICTAI'06* (pp. 599–608).

Petkova, D., & Croft, W. (2007). Proximity-based document representation for named entity retrieval. In *Proceedings of the 16th ACM conference on conference on information and knowledge management* (pp. 731–740). NY, USA: ACM New York.

Plachouras, V., He, B., & Ounis, I. (2005). University of Glasgow at TREC2004: Experiments in web, robust and terabyte tracks with Terrier. In *Proceedings of the 13th text retrieval conference (TREC)*.

Qin, T., Zhang, X., Tsai, M., Wang, D., Liu, T., & Li, H. (2008). Query-level loss functions for information retrieval. *Information Processing and Management, 44*(2), 838–855.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., & Gatford, M. (1996) Okapi at TREC-4. In *Proceedings of the 4th text retrieval conference (TREC)* (pp. 73–97).

Savoy, J., Le Calvé, A., & Vrajitoru, D. (1997). Report on the TREC-5 experiment: Data fusion and collection fusion. In *Proceedings of the 5th text retrieval conference (TREC)* (pp. 489–502). National Institute of Standards and Technology.

Serdyukov, P., & Hiemstra, D. (2008). Modeling documents as mixtures of persons for expert finding. In *Proceedings of 30th European conference on information retrieval* (Vol. 4956, p. 309). Springer.

Strohman, T., Metzler, D., Turtle, H., & Croft, W. (2004). Indri: A language model-based search engine for complex queries. In *Proceedings of the international conference on intelligence analysis*.

Vogt, C., & Cottrell, G. (1999). Fusion via a linear combination of scores. *Information Retrieval, 1*(3), 151–173.

Vogt, C., Cottrell, G., Belew, R., & Bartell, B. (1997). Using relevance to train a linear mixture of experts. In *Proceedings of the 5th text retrieval conference (TREC)* (pp. 503–515). National Institute of Standards and Technology.

Xu, J., & Li, H. (2007). Adarank: A boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 391–398). NY, USA: ACM New York.

Yan, R., & Hauptmann, A. (2006). 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* (pp. 324–331). NY, USA: ACM New York.

Yan, R., Yang, J., & Hauptmann, A. (2004). Learning query-class dependent weights in automatic video retrieval. In *Proceedings of the 12th annual ACM international conference on multimedia* (pp. 548–555). NY, USA: ACM New York.

Zhu, J., Song, D., Ruger, S., Eisenstadt, & M., Motta, E. (2006). The open university at TREC 2006 enterprise track expert search task. In *Proceedings of The 15th text retrieval conference (TREC 2006)*.