2008 Eighth IEEE International Conference on Data Mining

Formal Models for Expert Finding on DBLP Bibliography Data

Hongbo Deng, Irwin King, Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{hbdeng, king, lyu}@cse.cuhk.edu.hk

Abstract

Finding relevant experts in a specific field is often crucial for consulting, both in industry and in academia. The aim of this paper is to address the expert-finding task in a real world academic field. We present three models for expert finding based on the large-scale DBLP bibliography and Google Scholar for data supplementation. The first, a novel weighted language model, models an expert candidate based on the relevance and importance of associated documents by introducing a document prior probability, and achieves much better results than the basic language model. The second, a topic-based model, represents each candidate as a weighted sum of multiple topics, whilst the third, a hybrid model, combines the language model and the topic-based model. We evaluate our system using a benchmark dataset based on human relevance judgments of how well the expertise of proposed experts matches a query topic. Evaluation results show that our hybrid model outperforms other models in nearly all metrics.

1. Introduction

Expert finding has received increased interest in recent years since the advent of the expert search task in the TREC Enterprise track [22]. The task of expert finding is to come up with a ranked list of experts with relevant expertise in a given topic. The current developments in expert search are concentrated in the Enterprise corpora known as TREC2005 [8, 9] and TREC2006 [21]. They have provided a common platform for researchers to empirically assess methods and techniques devised for expert finding. However, little work has been done on methods of finding experts in any specific academic field, which is an important practical problem. Identification of the persons that have expertise on a specific academic topic could be of great value in many applications, for example, determining important experts for consultation by researchers embarking on a new research field, recommending panels of reviewers for state research grant applications [11], and assigning papers to reviewers automatically in a Peer-Review Process [17, 20].

As our approach is to deal with the expert-finding task in a real-world academic field, a key component is therefore the acquisition of a dataset replete with publications from which expertise can be accessed. The DBLP bibliography[1] is a good starting point for extracting the data needed for this application, as it contains more than 955,000 articles with over 574,000 authors from conferences and journals in the Computer Science field. In scientific research, the publications of a researcher could be assumed to be representative of his/her expertise [20]. One limitation of DBLP data is that each paper record only contains the title, which is too limited to calculate the relevance of papers to queries. To address this problem, Google Scholar [2] is utilized as a data supplement.

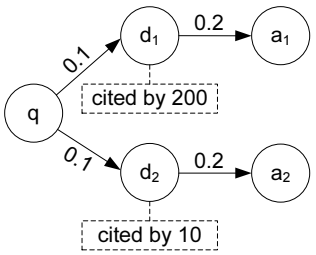
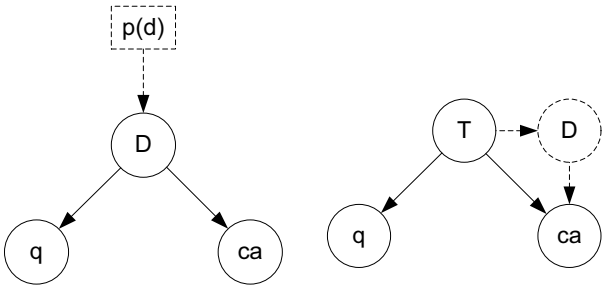


Figure 1. A query example with documents and authors.

Previous approaches in the TREC Enterprise Track [8, 9, 21] treat expert finding as an information retrieval task. One of the state-of-the-art approaches is based on a statistical language model to rank experts. The basic language model measures the relevance between a query and documents, then models the knowledge of an expert from the associated documents [3, 4]. An illustration of a query ex-

[1] http://www.informatik.uni-trier.de/~ley/db/

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.29



(a) Weighted model (b) Topic-based model

Figure 2. The models for expert finding.

ample is sketched in Figure 1. Here we suppose a query q has the same relevance probability (=0.1) to two documents d_1 and d_2; documents (d_1 and d_2) are associated with authors (a_1 and a_2) respectively (=0.2). In addition, d_1 is cited by 200 documents, while d_2 is cited by 10 documents. According to the basic language model and the above information, the author a_1 has the same expertise as the author a_2 given the query q. However, when considering the citation number, the document d_1, which has the higher citation number, would be more important than d_2. Therefore, intuitively, it is more reasonable that a_1 (the author of d_1) has a higher probability of being an expert than a_2 (the author of d_2) given the query q.

To address this problem, we introduce a novel weighted model to interpret the importance of the document d by introducing a prior probability $p(d)$, i.e., the prior probability of a document written by an expert, as shown in Figure 2 (a). Let D be the set of related documents, and ca be an expert candidate. In most existing work [3, 21], such as document-based model, this probability is ignored or assumed to be uniform. However, as shown in Section 6.2, a reasonable prior can help improve the retrieval accuracy. In this paper, we assume such a prior probability is related to the importance of the document. Specifically, we construct a weighted language model to take into consideration not only the relevance between a query and documents but also the impact of the documents. Then the expertise of the authors could be deduced based on the overall aggregation of the relevance and the document priors. Our evaluation results indicate that it is very important to consider the prior probability.

Moreover, motivated by the observation that researchers usually describe their expertise as a combination of several topics, we investigate a topic-based model to associate the query with the expert candidates. As shown in Figure 2 (b), each expert candidate is represented in terms of mixing proportions of multiple topics (denoted as T). So, the expertise given a query could be derived based on the proportionate aggregation of associated topics. Furthermore, we propose a hybrid model to combine the language model with the topic-based model. Experimental results show that a weighted language model can improve the performance significantly compared to the baseline language model, while the topic-based model achieves competitive results with language model. Finally, the evaluation results of the hybrid model show that it outperforms the language model and the topic-based model.

The main contribution of this paper is to propose an effective weighted language model, which introduces a document prior probability $p(d)$ to model the importance of the document written by an expert. Another contribution of this paper is that we investigate a topic-based model to interpret the expert finding task, and then integrate the language model with the topic-based model.

The rest of this paper is organized as follows. We briefly summarize related work on expertise and the topic model in Section 2. In Section 3 we provide detailed descriptions of the expertise modeling based on the language model. Advanced models including the topic-based model and the hybrid model for expertise retrieval are presented in Section 4. Next, in Section 5, we define the experimental setup of our methods. Experimental results are presented in Section 6. We conclude and discuss future work in Section 7.

2. Related Work

The inclusion of expert finding in the TREC Enterprise Track has resulted in a great deal of work in this area. There are two principal approaches to expert modeling: query-dependent and query-independent. In both cases the expert-finding system has to discover documents related to a person and estimate the probability of that person being an expert from the text [18]. A query-independent method [5, 15, 10] directly models the profile (builds a "virtual document") of a candidate based on all documents associated with the candidate and estimates the ranking score according to the profile in response to a user query. On the other hand, a query-dependent approach [3, 10] first ranks documents in the corpus given a query topic, and then find the associated candidates from the subset of retrieved documents.

Both methods have advantages and disadvantages [18]. In terms of data management, query-independent profiles can be significantly smaller in size than the original corpus. However, the contribution of each document in a profile cannot be measured individually, and as a result, this approach is less effective than other subsequent approaches. On the other hand, a query-dependent approach allows the application of advanced text modeling techniques in ranking individual documents.

Balog et al. [3] propose two language models in expert search and extensively compare the two methods. Their first model directly models the knowledge of an expert from as-

sociated documents, while their second model first locates documents on the query topic and then finds the associated experts. Their experimental results show that the second model outperforms the first model. Petkova and Croft [18] propose a hierarchical approach, based on a combination of the above first model and second model. In contrast, the probabilistic approach proposed by Cao et al. [8] uses a two-stage language model of combining relevance model and co-occurrence model. In [15], Macdonald and Ounis present yet another approach based on a voting model for expert search. Later, they apply query expansion in [16] to enhance the expert search. Nearly all of the work has been evaluated on the W3C collection [23]. Balog et al. [4] focus on expertise retrieval in an intranet that differs from the W3C setting.

The use of a topic model for information retrieval tasks is described in Wei and Croft [24]. The authors find that interpolations between Dirichlet smoothed language models and topic models show improvements in retrieval performance above language models by themselves. Probabilistic latent semantic indexing (pLSI) is a widely used document model [12, 6]. Furthermore, Mimno and McCallum [17] propose the Author-Persona-Topic model to model the expertise of a person. However, computational complexity is often a big concern for topic models. Our proposed topic-based model can be simplified using a predefined topic set.

Despite all this work in expert finding, little work has been done in a specific academic domain, in terms of retrieving experts given the topic. Recently, Li et al. [14] build an academic expertise oriented search service, including expert finding based on the DBLP bibliography. They propose a relevancy propagation-based algorithm using the co-authorship network for expert finding. In this paper we focus on expert finding in a real world academic field based on the DBLP bibliography.

3. Statistical Language Model

In this section we detail the expert finding models and introduce a set of baseline approaches based on statistical language modeling; later, in Section 4, we turn our attention to advanced modeling approaches, including the topic-based model and the hybrid model.

3.1 Basic Language Model

Using language models for information retrieval has been studied extensively in recent years [13, 19, 25, 26]. The basic idea of these approaches is to estimate a language model for each document, and then rank documents by the likelihood of their matching the query according to the estimated language model. Several different methods have been applied to compute the query likelihood, i.e., the probability of generating a query given the observation of a document.

In [3], Balog et al. formulate the problem of identifying experts for a given topic using a generative probabilistic model: what is the probability of a candidate ca being an expert given the query topic q? Thus, the task is to determine $p(ca|q)$, and rank candidates ca according to this probability. There are no restrictions on the form of the query topic q, which could consist of any terms or concepts; for instance, "data mining" is a query to search experts who have expertise on the query topic "data mining". Using Bayes' theorem, the probability can be formulated as follows:

$$p(ca|q) = \frac{p(ca, q)}{p(q)}, \tag{1}$$

where $p(ca, q)$ is the joint probability of a candidate and a query, $p(q)$ is the probability of a query. Since $p(q)$ is a constant, it can be ignored for ranking purposes. The probability $p(ca|q)$ can be reformulated to estimate the joint probability $p(ca, q)$. The basic language model used to estimate the probability $p_l(ca, q)$ can be defined as follows:

$$\begin{aligned} p_l(ca, q) &= \sum_{d \in D} p(d) p(ca, q|d) \\ &= \sum_{d \in D} p(d) p(q|d) p(ca|d, q), \end{aligned} \tag{2}$$

where $p(d)$ is the prior probability of a document, and the supporting documents D act as a "bridge" to connect q and ca. Under this model, the process of finding an expert is as follows: given a collection of documents ranked according to the query, we examine each document relevant to the query, and then we note the authors associated with that document. Here, the process is taken to the extreme where we consider all documents in the collection.

To determine the probability of a query given a document, we infer a document language model θ_d for each document,

$$p(q|\theta_d) = \prod_{t \in q} p(t|\theta_d)^{n(t,q)}, \tag{3}$$

where $p(t|\theta_d)$ is the maximum likelihood estimate of the term in a document d, and $n(t, q)$ is the number of times that term t occurs in query q. This model is drawn from Balog et al. [3]. The likelihood of a query q consisting of some number of terms t for a document d under a language model with Jelinek-Mercer smoothing [26] is

$$p(t|\theta_d) = (1 - \lambda) p(t|d) + \lambda p(t). \tag{4}$$

We follow Balog et al. in setting $\lambda = 0.5$.

Suppose that we make the assumption that the candidate ca is conditionally independent of the query q given a document d; that is

$$p(ca|d, q) = p(ca|d). \tag{5}$$

In our setting it is reasonable to assume that candidate ca has knowledge about the topic described in the document d if candidate ca is an author of document d. Now, in the case of a multi-author paper, one author with many co-authors may have less association $p(ca|d)$ on average than a sole author. To account for this effect, we weight the association inversely according to the number of co-authors as follows. Suppose a document has n authors in total, we assume that each author has the same knowledge about the topics described in the document,

$$p(ca|d) = \begin{cases} \frac{1}{n_d}, & (ca \text{ is the author of } d) \\ 0, & (\text{otherwise}) \end{cases} \quad (6)$$

where n_d is the number of authors, and $p(ca|d)$ is used to measure the document-candidate association.

The document priors $p(d)$ are generally assumed to be uniform and thus will not influence the ranking. The final estimation of the baseline language model is obtained by substituting $p(q|\theta_d)$ for $p(q|d)$ into Eq. 2,

$$p_l(ca, q) \stackrel{rank}{=} \sum_{d \in D} \left\{ \prod_{t \in q} (p(t|\theta_d))^{n(t,q)} \right\} p(ca|d), \quad (7)$$

where $\stackrel{rank}{=}$ means "equivalence for ranking the candidates". We refer to this method as a baseline language model for expert finding.

3.2 Weighted Language Model

The language model described above calculates the relevance between a query and a document, but it ignores the prior of the document. As shown in Figure 1, suppose there are only two documents d_1 and d_2, d_1 with one author a_1 and d_2 with one author a_2; both documents have similar contents, i.e., the query likelihoods are almost the same ($p(q|d_1) = p(q|d_2)$), but the two documents have different importance, $I(d_1) > I(d_2)$. Which document is more reasonable to rank to the top? Which author has the higher probability of being an expert given the query topic? Obviously, we would prefer to rank the more important one (d_1) at the top, and author a_1 would have the higher probability of being an expert than author a_2 on topic q based on the assumption that the more important document has more weight. To the best of our knowledge, the language models currently do not take this factor into account. We introduce a weight factor w_d to denote the importance of the document, which, theoretically, can be interpreted as being proportional to the document prior $p(d)$,

$$p(d) = \frac{w_d}{C} \propto w_d, \quad (8)$$

where C ($= \sum_{d \in D} w_d$) is a constant normalization factor.

For our model, the weight factor is estimated using the citation number, and transformed using two logarithm functions: the common logarithm (B2), and the natural logarithm (B3). We can see that this is exactly the method of the basic language model when the uniform weight w_d is set to 1 (B1). Three different methods to measure the weight are defined as follows,

$$w_d = \begin{cases} 1, & (B1) \\ \log(10 + c_d), & (B2) \\ \ln(e + c_d), & (B3) \end{cases} \quad (9)$$

where c_d ($c_d \geq 0$) is the citation number of the document d, and the constants 10 and e are used to guarantee the weight factor not to be less than 1. The citation numbers are obtained from Google Scholar [2].

The final estimation of the weighted language model is

$$p_l(q, ca) \stackrel{rank}{=} \sum_{d \in D} w_d \left\{ \prod_{t \in q} (p(t|\theta_d))^{n(t,q)} \right\} p(ca|d). \quad (10)$$

In Section 6.2, we compare the performance of weighted language models with different weighting methods, namely $B1$, $B2$ and $B3$.

4. Advanced Models

Now that our language modeling techniques have been developed for expertise retrieval, we proceed to introduce our topic-based model. Also, in this section, a hybrid model is presented which combines the language model with the topic-based model.

4.1 Topic-based Model

4.1.1 Model Form

In this approach, as illustrated by the model in Figure 2 (b), each candidate is represented as a weighted sum of multiple topics, and there is an implicit relation between the query and the topic z in terms of the probability $p(q|z)$.

In our topic-based model, a candidate ca and a query q are conditionally independent given a latent topic z:

$$\begin{aligned} p_t(q, ca) &= \sum_{z \in Z} p(q|z, ca)p(ca|z)p(z), \\ &= \sum_{z \in Z} p(q|z)p(z, ca), \end{aligned} \quad (11)$$

where $p(z, ca)$ is the joint probability of the topic z and the candidate ca, and $p(q|z)$ represents the probability of a query q generated by the topic z. Computational complexity is often a big concern for topic models, especially when the dataset is large-scale and a great number of latent variables

are used. Our topic-based model can therefore be simplified if a well-defined topic set is available, and it can be viewed as query expansion.

If such a well-defined topic set is available, the probability $p(z, ca)$ could be estimated using the method as described in Section 3. In addition, we need to associate the well-defined topic z with the query topic q. Therefore, given a well-defined topic, we can interpret this topic by collecting a number of most relevant papers as the supporting representation of the topic z. We denote this as θ_z, then substitute θ_z for z into Eq. 11. Meanwhile, the probability $p(q|\theta_z)$ could be measured using Eq. 3, which indicates the correlation between the query q and the topic z, and the higher probability corresponds to the stronger connection. Thus, the probability can also be regarded as the similarity between each other. Now the remaining problem is transformed to find the most relevant or similar topics to the original query.

4.1.2 Topic Selection Algorithm

The challenge now is to consider what similar topics the candidate would satisfy, and then use the entire subset of topics similar to the original query to measure the joint probability of a query and a candidate. We advocate three methods for selecting the similar topics (the subset of topics) and estimating this probability.

The first method is conceptually simpler, and assumes that those topics are independent. Let $Z = \{z_1, z_2, ..., z_n\}$ be the set of all predefined topics. Then, one natural way to select topics similar to the query is to calculate the similarity score $p(q|\theta_z)$ between the query and topics as the ranking function, and use the top K ranked topics as the similar topics. We denote this method as T1.

By definition of Eq. 11, the topic model builds on a kind of independent relevance assumption: there is no spillover of expertise across predefined topics. This assumption rarely holds in reality. Intuitively, it is desirable for the selected similar topics to include topics from many different subtopics and undesirable that they include many topics that redundantly cover the same subtopics. However, the first method T1 may select a subset of topics with high redundancy, which may induce expertise topic drift in the topic-based model. To take into account of the topic dependence, we consider another method of selecting the subset of topics, which approximately satisfies the independent relevance assumption, from all the predefined topics.

To obtain such similar topics, a greedy algorithm is designed to select topics one by one according to the given query. Suppose we have selected several topics $z_1, ..., z_{i-1}$, the next topic z_i should cover many subtopics not covered by the previous topics, and few of the subtopics covered by the previous topics. It can be formulated using a conditional probability function $value(q|z_i; z_1, ..., z_{i-1})$, i.e., to quantify the novelty and penalize the redundancy of a topic z_i for rank i. The detailed algorithm is shown in Algorithm 1. In each round, the algorithm tries to exhaustively search for the topic with the maximum value function. In the course of K rounds, we get K similar topics. Note that the algorithm will stop when the value function is less than or equal to 0.

Since it is difficult to represent the value function explicitly, some kind of approximation is necessary in practice. An approach for approximating the value function is defined as follows:

$$value(q|z_i; z_1, ..., z_{i-1}) \approx p(q|z_i) - \max_{j<i} p(z_i|z_j), \quad (12)$$

where $\max_{j<i} p(z_i|z_j)$ is the maximum similarity between z_i and the previously selected similar topics. We denote this method as **T2**. Another way to approximate the value function is formulated as follows:

$$value(q|z_i; z_1, ..., z_{i-1}) \approx p(q|z_i) - \sum_{j<i} p(q|z_j)p(z_j|z_i). \quad (13)$$

We denote the method using Eq. 13 as **T3**. Once we obtain the subset of topics, then we use Eq. 11 to calculate the joint probability $p_t(q, ca)$.

Algorithm 1 Greedy Selection Algorithm

Inputs: Predefined topic set Z, select topic size K

1: **for** $i = 1 \dots K$ **do**
2: Select a well-defined topic z_i from unselected topic set Z to maximize the value of

$$z_i = \arg\max_{z_i \in Z}(value(q|z_i; z_1, ..., z_{i-1})).$$

3: Prerequisite: $value(q|z_i; z_1, ..., z_{i-1}) > 0$, otherwise exit the loop.
4: Update the topic set:

$$Z = Z - \{z_i\}.$$

5: **end for**

Return the topics $\{z_1, z_2, ..., z_K\}$.

4.2 Hybrid Model

To improve the performance, a hybrid model is utilized to aggregate the advantage of the language model and the topic-based model. Consider the probability of a candidate ca being an expert given the query topic q: this can be modeled by interpolating between the language model $p_l(q, ca)$ and the topic-based model $p_t(q, ca)$, as follows:

$$p_h(q, ca) = \mu p_l(q, ca) + (1 - \mu)p_t(q, ca). \quad (14)$$

In Section 6.4 we compare the performance of the hybrid model with the pure language model and the topic-based model, and demonstrate that the hybrid model can provide more accurate results than the pure approaches.

5. Experimental Setup

In the following experiments we compare the three different expert finding models through an empirical evaluation. In this section we define the experimental setup, while the evaluation results are presented in Section 6.

We have defined the following task: given a query and a set of expert candidates, the system has to retrieve a list of experts that have expertise in the given area. In the rest of this section, we introduce the DBLP and topic collection, the assessments and evaluation metrics.

5.1 DBLP and Topic Collection

A key aspect of finding experts from bibliographic data is therefore the acquisition of a dataset replete with publications from which expertise can be derived. As of November 2007, DBLP XML records contain over 955,000 articles related to Computer Science, originally published in conferences, journals, books etc., adding up to 414.5MB. In total we gather more than 574,000 author names from DBLP XML records, each of whom can be an expert candidate. Although DBLP is a good starting point for obtaining expert candidates and publications, several challenges exist due to its limitations. One limitation is that each DBLP record provides the paper title without the abstract and index terms. The information provided by the title is too limited to represent the paper; some more expanded information is required. Generally, the abstract and index terms are useful to represent the paper for estimating the probability of a query or topic given the paper.

To obtain the abstract and index terms for each DBLP record, one natural way is to fetch them automatically from digital libraries such as ACM, IEEE, Springer, etc. We note, however, that it is very hard to obtain the complete metadata (the abstracts and index terms of publications) for all the DBLP records. Thus Google Scholar is utilized for data supplementation as shown in Figure 3: for a document d, we use the title as the query to search in Google Scholar and select the top 10 returned records which are most relevant to the query title; next, these records combined with the publication title are viewed as the representation of the publication d. The metadata (HTML pages) crawled from Google Scholar is up to 20GB. This process is done automatically by a crawler and a parser, and the citation number of the publication d in Google Scholar is obtained at the same time. The total number of valid papers after this process is 953,774, and the number of valid authors is 574,369.

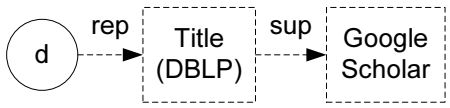


Figure 3. The representation of a document.

For the topic-based model, an important task is to collect the predefined topics related to Computer Science. From eventseer.net [1], an interesting website that tracks upcoming Computer Science research events, one can obtain an updated repository of 2,498 well-defined topics. Table 1 shows a snippet of topics from eventseer.net. Working with this list of topics, we also use the method based on Google Scholar to crawl the top 100 returned records as the supporting representation of each topic. The statistics of DBLP and the topic collection are shown in Table 2.

Table 1. Example topics from eventseer.net.

Example topics	
Machine architecture	Magnetic field
Machine learning	Magnetic resonance
Machine learning algorithms	Magnetic resonance images
Machine learning and data mining	Main memory
Machine learning applications	Maintenance and evolution
Machine scheduling	Maintenance of competence
Machine translation	Maintenance of data warehouses
Machine vision	Maintenance of semantic mappings
Mac layer	Maintenance, reuse and evolution
Mac protocol	Management framework

Table 2. Statistics of DBLP and the topic collection.

Property	#of entities
DBLP:no_of_pub	953,774
DBLP:no_of_author	574,369
Topic:no_of_topic	2,498

5.2 Assessments

It is difficult to evaluate the quality of query/expert relevance rankings due to the scarcity of data that can be examined publicly. The ground truth is manually created through the method of pooled relevance judgments together with human judgments. For each query, the top authors from

Table 3. Benchmark dataset of 7 topics.

Topic	#Expert
Information Extraction	20
Intelligent Agents	29
Machine Learning	42
Natural Language Processing	43
Planning	34
Semantic Web	45
Support Vector Machine	31

the computer science bibliography search engines (such as CiteSeer[2], Libra[3], and Rexa[4]) and the committees of the top conferences in the topic were taken to construct the pool. Some researchers were then asked to assess each of the recommended candidates in context of the query. To help them in their task, those researchers were presented with publications and a description relating to each author. They could access and find additional content directly on a search engine when needed.

Such a benchmark dataset with expert lists (for expert finding) has been collected in Tsinghua university [27]. Their assessments were carried out mainly in terms of how many publications an expert candidate has published, how many publications are related to the given query, how many top conference papers he/she has published, and what distinguished awards he/she has been awarded. Four grade scores (3, 2, 1, and 0) were assigned respectively representing top expert, expert, marginal expert, and not expert. Finally, the judgment scores (at levels 3 and 2) were averaged to construct the final ground truth. The data set contains 7 query topics and creates 7 expert lists. Table 3 shows the details of the dataset.

5.3 Evaluation Metrics

For the evaluation of the task, we adopted three metrics that capture different aspects of the performance of our proposed models.

Precision at rank n (P@n) Precision at rank n measures the relevance of the top n results of the retrieved list with respect to a given query topic. R-precision (R-prec) is defined as the precision at rank R where R is the number of relevant candidates for the given query topic. We report the precision P@10, P@20, P@30, and R-prec.

$$P@n = \frac{\text{\# relevant candidates in top } n \text{ results}}{n} \quad (15)$$

[2]http://citeseer.ist.psu.edu/
[3]http://libra.msra.cn/
[4]http://rexa.info/

Mean Average Precision (MAP) For a single query, average precision (AP) is defined as the average of the P@n values for all relevant documents:

$$AP = \frac{\sum_{n=1}^{N}(P@n * \text{rel}(n))}{R} \quad (16)$$

where n is the rank, N the number retrieved, and rel(n) is a binary function indicating the relevance of a given rank. MAP is the mean value of the average precisions computed for several queries.

Bpref Bpref [7] is the score function of the number of non-relevant candidates:

$$\text{bpref} = \frac{1}{R}\sum_{r=1}^{N}(1 - \frac{\#n \text{ ranked higher than } r}{R}) \quad (17)$$

where r is a relevant candidate and n is a member of the first R candidates judged nonrelevant as retrieved by the system.

6. Evaluation Results

The presentation of the evaluation results is organized in the following five subsections. First we evaluate the effectiveness of the representation of each publication using Google Scholar. Then we report the results for the language models and compare the three weighting methods in Section 6.2. In Section 6.3, we examine the performance of the different topic-based models. Section 6.4 discusses the results for the hybrid models, in comparison to the pure language models and topic models. Finally, we compare our models with other methods. The evaluation results shown in this section are the average results.

6.1 Preliminary Experiments

As described in Section 5.1, the DBLP records only contain the publication titles. We present a new and effective representation for a publication based on Google Scholar. In order to compare the performance of the two representations, we set up two corpora for evaluation. One corpus (Title) is collected only using the publication title, while the other corpus (GS) is built based on the supplemental representation using Google Scholar. Preliminary experiments are performed on these two corpora using the basic language model (B1). The comparison results are reported in Table 4. It is clear that the results of "GS" are much better than those of "Title", which indicates that it is more effective to represent publications using Google Scholar as a data supplement. Thus the "GS" corpus is used in the following parts.

Table 4. Evaluation results on two corpora (%).

	P@10	P@20	P@30	R-prec	MAP	bpref
Title	57.14	42.86	40.00	38.65	22.92	30.05
GS	61.43	50.71	42.38	43.21	30.38	35.95

Table 5. Evaluation results of language models using different weighting methods (%). Best scores are in boldface.

	P@10	P@20	P@30	R-prec	MAP	bpref
B1	61.43	50.71	42.38	43.21	30.38	35.95
B2	65.71	53.57	**48.10**	44.42	32.03	37.30
B3	**68.57**	**57.86**	46.19	**44.48**	**32.39**	**37.70**

Table 6. Evaluation results of topic-based models using different numbers of similar topics (%).

	P@10	P@20	P@30	R-prec	MAP	bpref
Model: T1						
K=5	62.86	52.14	43.33	40.98	29.02	34.39
K=10	62.86	50.71	43.81	39.56	28.19	33.46
K=20	58.57	48.57	42.38	37.82	26.50	31.87
K=40	57.14	47.14	39.05	37.02	24.83	29.83
K=100	50.00	40.71	36.19	33.32	21.09	26.61
Model: T2						
K=5	68.57	55.71	46.19	43.40	31.45	37.00
K=10	70.00	55.71	46.19	43.40	31.51	37.01
K=20=40=100	the same results as K=10					
Model: T3						
K=5	70.00	56.43	46.19	44.11	31.86	37.39
K=10=20=40=100	the same results as K=5					

6.2 Language Models

In this subsection we evaluate the performance of the language models and compare the three different weighting methods. Table 5 shows the results for the different methods on the test collection, where B1 represents the baseline method with uniform weight $w_d = 1$, B2 is the method with the common logarithm weight $w_d = \log(10+n_d)$ and B3 is the method with natural logarithm weight $w_d = \ln(e+n_d)$.

First, we inspect the absolute performance of the methods. For the precision P@10, the basic language model B1 only achieves 61.43%, and the weighted language models B2 and B3 can enhance the precision significantly to 65.71% and 68.57%. For the mean average precision (MAP), we measure a precision of 32.39% for B3, and 30.38% for B1, which indicates that B3 improves the MAP measurement by 6.7%. When looking at the overall performance, we observe that weighted language models B3 and B2 outperform the basic language model B1 on all the metrics from P@10 to bpref. Comparing the two weighted language models, method B3 is better than method B2 in most cases.

According to the experimental results, we can argue that it is very important to consider the prior probability of the document; our weighted language model performs very well and achieves much better performance than the basic language model. Based on the outcomes of our experiments, we use the weighting method B3 in the following parts of the section.

6.3 Topic-based Models

We now turn our attention to the performance of our topic-based models. In this subsection, we evaluate and compare the three topic-based models introduced in Section 4.1, varying the number of topics (K) from 5 to 100. For T1, K denotes the top ranked topics. However, in T2 and T3, K represents the number of selected topics in Algorithm 1, and the algorithm may stop before completing K rounds. In this event, we denote round i as a cut-off point, and the result in this point may be close to the best performance in a sense. Table 6 shows the detailed results using different values for K.

A quick scan of Table 6 reveals that T2 and T3 always outperform T1 for all settings. Figure 4 compares the performance of the three topic-based models with different numbers of topics, using different metrics. For T1, increasing the number of topics was not of benefit to the performance. In fact, the results of T1 are inversely related to the number of topics, and the best results were achieved when K was equal to 5. In contrast, for T2, we witness improvements with increasing number of topics in some cases, and the best results were achieved when K was 10. In terms of the precision P@10, T2 and T3 achieved higher performance than the best language model B3.

Importantly, the number of topics will be cut off automatically when K is larger than 10 in T2. For T3, the cut-off point is 5. The differences between T1 and T2/T3 are significant for different numbers of similar topics. As expected, it may mean that T2 and T3 perform better since it reduces the redundancy between the selected similar topics.

6.4 Hybrid Models

In this section, we evaluate the hybrid model by tuning μ from 0 to 1 with increments of 0.1. A hybrid model can

Table 7. Evaluation results of hybrid models (%). Best scores are in boldface.

	P@10	P@20	P@30	R-prec	MAP	bpref
Hybrid Model H2: (B3 + T2 with K=10)						
B3	68.57	**57.86**	**46.19**	**44.48**	32.39	37.70
T2	70.00	55.71	**46.19**	43.40	31.51	37.01
H2	**71.43**	**57.86**	**46.19**	**44.82**	**32.60**	**37.98**

Table 8. Example results of the hybrid model showing the top five experts for several queries, based on the DBLP dataset.

Q1: Boosting	Q2: Data Mining
Robert E. Schapire	Jiawei Han
Yoav Freund	Mohammed Javeed Zaki
Yoram Singer	Rakesh Agrawal
Manfred K. Warmuth	Heikki Mannila
Nader H. Bshouty	Philip S. Yu
Q3: Information Extraction	**Q4: Semantic Web**
Ellen Riloff	Dieter Fensel
Dayne Freitag	James A. Hendler
Stephen Soderland	Katia P. Sycara
Raymond J. Mooney	Amit P. Sheth
Andrew McCallum	Ian Horrocks
Q5: Support Vector Machine	**Q6: Computer Vision**
Bernhard Schölkopf	Azriel Rosenfeld
Vladimir Vapnik	Robert M. Haralick
Alex J. Smola	Michael Brady
Ingo Steinwart	Dana H. Ballard
Thorsten Joachims	Thomas S. Huang

consist of any language models and topic-based models. We restrict ourselves to the combination of the best performing language model B3 and the topic-based model T2 with K equal to 10, namely H2. For all the measure metrics, H2 returns the highest performance when $\mu = 0.6$. Table 7 reports detailed results of the hybrid model H2 with weight $\mu = 0.6$ compared to B3 and T2. The improvement of the hybrid model H2 is relatively small, as the performances of the model T2 and the model B3 are very similar. However, for the top 10 precision (P@10), H2 outperforms both T2 and B3, improving the precision from 68.57% to 71.43%. In general, the evaluation results show that our hybrid model outperforms the pure language model and topic-based model in most of the metrics.

For illustration, we show six examples of the top 5 experts in Table 8, where the query samples are "Boosting", "Data Mining", "Information Extraction", "Semantic Web", "Support Vector Machine", and "Computer Vision".

Table 9. Evaluation results of our language models and the method TS (%). Best scores are in boldface.

	P@10	P@20	P@30	R-prec	MAP	bpref
TS	48.00	40.40	36.15	-	11.03	16.11
B1	53.85	43.46	39.74	22.40	11.36	17.16
B2	59.23	50.77	43.33	23.93	13.33	18.94
B3	**60.77**	**51.54**	**43.85**	**24.83**	**13.92**	**19.67**

6.5 Comparison to Other Systems

To compare with the approaches proposed by [14, 27], we set up the experiments with the same benchmark dataset, which contains 13 query topics and corresponding expert lists. We denote their method as **TS**. Table 9 shows the evaluation results of our language models and the method TS reported by Tsinghua. Clearly, our results are much better than the method TS for all the metrics. For our language models, we observe that method B3 outperforms B2, and B2 outperforms B1. These results are consistent with the results shown in Section 6.2.

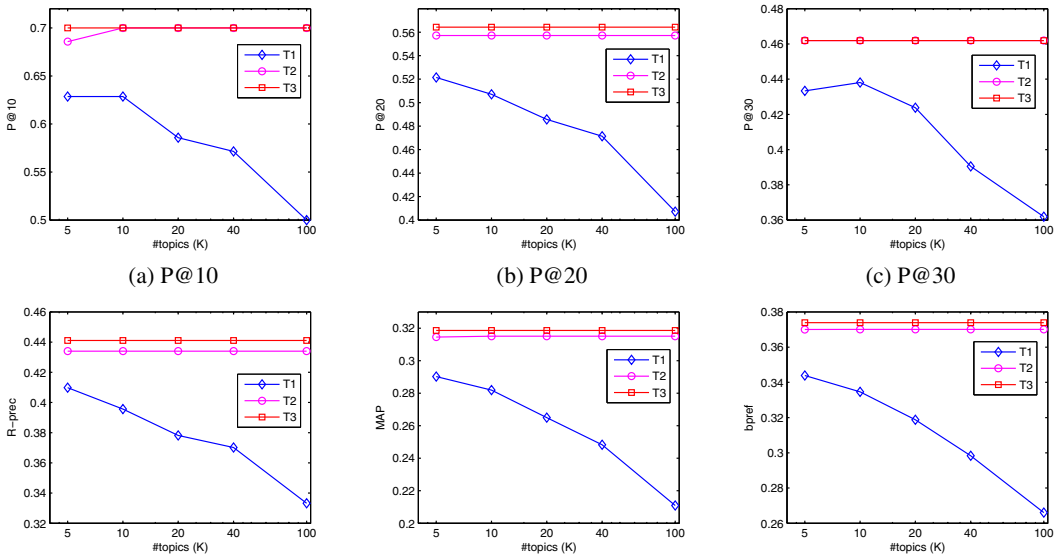
7. Conclusions and Future Work

We presented our three expert-finding models, whose purpose is to retrieve experts in specific academic domains based on the DBLP bibliography and Google Scholar for data supplementation. Our models include the statistical language model, the topic-based model and the hybrid model. More specifically, we proposed a weighted language model and a topic-based model with predefined topics. We have shown in our evaluation results that, in general, the weighted language model improves the performance significantly compared to the baseline language model. In terms of the topic-based model, we have proposed three methods to search for experts. As expected, the topic-based models T2 and T3, which reduce the redundancy within the subset of topics, perform much better than T1. Finally, the evaluation results of our hybrid model show that it outperforms the pure language model and the topic-based model.

Our current expert-finding approaches in the DBLP dataset only consider the publications of the experts. To further improve the performance of our methods, we plan to take into account other types of information in future work, such as profiles of the researchers and social information.

8 Acknowledgments

This work is fully supported by two grants from the Research Grants Council of the Hong Kong Special Admin-



(a) P@10 (b) P@20 (c) P@30

(d) R-prec (e) MAP (f) bpref

Figure 4. Comparison of the three topic-based models with different numbers of topics.

istrative Region, China (Project No. CUHK 4125/07E and Project No. CUHK4150/07E).

References

[1] Eventseer.net. URL:http://eventseer.net/topic/.

[2] Google Scholar. URL:http://scholar.google.com/.

[3] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR*, pages 43–50, 2006.

[4] K. Balog, T. Bogers, L. Azzopardi, M. de Rijke, and A. van den Bosch. Broad expertise retrieval in sparse data environments. In *SIGIR*, pages 551–558, 2007.

[5] K. Balog and M. de Rijke. Determining expert profiles (with an application to expert finding). In *IJCAI*, pages 2657–2662, 2007.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[7] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR*, pages 25–32, 2004.

[8] Y. Cao, J. Liu, S. Bao, and H. Li. Research on expert search at enterprise track of trec 2005. In *Proceedings of TREC 2005*, 2005.

[9] N. Craswell, I. Soboroff, and A. de Vries. Overview of the trec-2005 enterprise track. In *Proceedings of TREC 2005*.

[10] H. Fang and C. Zhai. Probabilistic models for expert finding. In *ECIR*, pages 418–430, 2007.

[11] S. Hettich and M. J. Pazzani. Mining for proposal reviewers: lessons learned at the national science foundation. In *KDD*, pages 862–871, 2006.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[13] R. Jin, A. G. Hauptmann, and C. Zhai. Title language model for information retrieval. In *SIGIR*, pages 42–48, 2002.

[14] J.-Z. Li, J. Tang, J. Zhang, Q. Luo, Y. Liu, and M. Hong. Eos: expertise oriented search using social networks. In *WWW*, pages 1271–1272, 2007.

[15] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *CIKM*, pages 387–396, 2006.

[16] C. Macdonald and I. Ounis. Expertise drift and query expansion in expert search. In *CIKM*, pages 341–350, 2007.

[17] D. M. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *KDD*, pages 500–509, 2007.

[18] D. Petkova and W. B. Croft. Hierarchical language models for expert finding in enterprise corpora. In *ICTAI*, pages 599–608, 2006.

[19] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, 1998.

[20] M. A. Rodriguez and J. Bollen. An algorithm to determine peer-reviewers. *CoRR*, abs/cs/0605112, 2006.

[21] I. Soboroff, A. de Vries, and N. Craswell. Overview of the trec-2006 enterprise track. In *Proceedings of TREC 2006*.

[22] TREC. Enterprise track, 2005. URL: http://www.ins.cwi.nl/projects/trec-ent/wiki/.

[23] W3C. The W3C test collection, 2005. URL: http://research.microsoft.com/users/nickcr/w3c-summary.html.

[24] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185, 2006.

[25] C. Zhai and J. D. Lafferty. Two-stage language models for information retrieval. In *SIGIR*, pages 49–56, 2002.

[26] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[27] J. Zhang, J. Tang, and J.-Z. Li. Expert finding in a social network. In *DASFAA*, pages 1066–1069, 2007.