Enhanced Models for Expertise Retrieval Using Community-Aware Strategies

Hongbo Deng, Irwin King, *Senior Member, IEEE*, and Michael R. Lyu, *Fellow, IEEE*

***Abstract*—Expertise retrieval, whose task is to suggest people with relevant expertise on the topic of interest, has received increasing interest in recent years. One of the issues is that previous algorithms mainly consider the documents associated with the experts while ignoring the community information that is affiliated with the documents and the experts. Motivated by the observation that communities could provide valuable insight and distinctive information, we investigate and develop two community-aware strategies to enhance expertise retrieval. We first propose a new smoothing method using the community context for statistical language modeling, which is employed to identify the most relevant documents so as to boost the performance of expertise retrieval in the document-based model. Furthermore, we propose a query-sensitive AuthorRank to model the authors' authorities based on the community coauthorship networks and develop an adaptive ranking refinement method to enhance expertise retrieval. Experimental results demonstrate the effectiveness and robustness of both community-aware strategies. Moreover, the improvements made in the enhanced models are significant and consistent.**

***Index Terms*—Community-aware strategy, expertise retrieval, language model, query-sensitive AuthorRank.**

I. Introduction

With the development of information retrieval (IR) techniques, many research efforts in this field have been made to address high-level IR and not just traditional document retrieval, such as entity retrieval [1], [2] and expertise retrieval [3]. Expertise retrieval has received increasing interest since the introduction of an expert finding task in the Text Retrieval Conference (TREC) 2005 [4], [5]. The task of expertise retrieval is to identify a set of persons with relevant expertise for the given query. Traditionally, the expertise of a candidate is characterized based on the documents that have been associated with the candidate. One of the state-of-the-art approaches [6], [7] is the document-based model to estimate the weighted sum of retrieval scores of all documents related to an expert candidate as a measure of candidate's expertise. However, previous methods mainly consider the documents associated with the experts, while ignoring the community information affiliated with the documents and the experts, such as the community context information and the community social information. Therefore, how to utilize the community-based information to enhance expertise retrieval becomes an interesting and challenging problem.

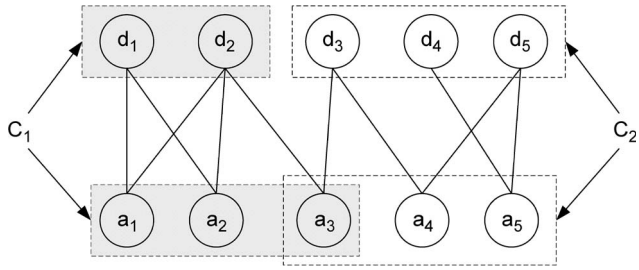

Fig. 1. Example graph with two communities.

Given a set of documents and their authors, it is possible and often desirable to discover and infer community information, which contains a number of documents and authors for each community. Some existing studies have been conducted about how to discover communities [8], [9], but this is not the focus of our work: Here, we focus on the problem of enhancing expertise retrieval with community information and suppose the community information already exists. As our approach is to deal with the expert-finding task in a real-world academic domain, it is reasonable to assume that the academic communities have been formed automatically in the form of conferences and journals, in which the researchers publish their papers and exchange their ideas with each other.

The community could provide valuable information for its documents and authors. Let us take Fig. 1 as an example. There are two communities which include five documents associated with five authors. The edge between a document and an author means the document is written by the author. We assume that each document d_i can only belong to one community C_k and each author a_j of the document is affiliated with the corresponding community C_k. In this example, d_1 and d_2 belong to community C_1, and meanwhile, d_3, d_4, and d_5 form community C_2. The authors of the documents a_1, a_2, and a_3 are affiliated with community C_1, and a_3, a_4, and a_5 with community C_2, so a single author may belong to multiple communities. For each community, there is a pair of distributions: one over documents [e.g., $p(t|C_d)$ in (10)] and the other over authors [e.g., $p(a|C_k)$ in (15)]. With such community-based information, the community can be represented from two different perspectives to obtain the community context (text information) based on papers and the community coauthorship network based on authors.

In this paper, we propose two community-aware strategies to enhance expertise retrieval. The first one is the community-based smoothing method for statistical language modeling,

Manuscript received October 13, 2009; revised August 31, 2010 and April 6, 2011; accepted June 28, 2011. Date of publication August 8, 2011; date of current version December 7, 2011. This work was supported by the Research Grants Council of Hong Kong under two grants (Projects CUHK 413210 and CUHK 415410). This paper was recommended by Associate Editor M. Huber.

H. Deng is with the Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, IL 61801-2302 USA (e-mail: hbdeng@uiuc.edu).

I. King is with AT&T Labs Research, San Francisco, CA 94105 USA, and also with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: king@cse.cuhk.edu.hk).

M. R. Lyu is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: lyu@cse.cuhk.edu.hk).

Digital Object Identifier 10.1109/TSMCB.2011.2161980

which is employed to identify the most relevant documents so as to boost the performance of expertise retrieval in the document-based model. The smoothing method is an important characteristic of the language model for computing the relevance score. In previous approaches [6], [7], the document language model is smoothed by the whole collection language model, which smooths each word equally in all the documents while ignoring their different community information. However, we argue that the community context provides more valuable and distinctive information for its documents than the whole collection. For example, as shown in Fig. 1, suppose C_1 denotes a "machine learning" community and C_2 denotes an "IR" community. Thus, it is likely to contain a higher proportion of words related to "machine learning" in the context of C_1 than the whole collection, and meanwhile, there would be a higher proportion of words related to IR in the context of C_2 than the whole collection. Generally, a document will somewhat share much more common information with its community rather than the whole collection, while different communities can be used to distinguish the documents. This observation motivates us to conduct the novel smoothing method using the community context.

The second strategy is developed to boost the document-based model using the community-sensitive authorities. More specifically, we propose a query-sensitive AuthorRank to model the authors' authorities based on the coauthorship networks and develop an adaptive ranking refinement method to aggregate the ranking results. Intuitively, experts usually have high authorities in some communities, which reflect their general and high-level expertise in some aspects. In contrast, the document-based model reflects more specific and detailed aspects for expertise retrieval, as it measures the contribution of each document individually. From this point of view, the community-sensitive authorities should be taken into consideration along with the document-based model for expertise retrieval, which is referred to as the enhanced model.

To illustrate our methodology, we apply the proposed methods to the expert finding task using the Digital Bibliography and Library Project (DBLP) bibliography data.[1] The evaluation of expert finding performance in such a large data collection is very challenging due to the scarcity of ground truth that can be examined publicly. Furthermore, it is unclear how to define the appropriate metrics to measure the quality of the retrieved experts given the scale involved and the impracticality of obtaining expert ratings for all authors. Following traditional IR practice, we evaluate our methods using expert ratings from a sample of authors, over a test set of queries. Experimental results demonstrate the effectiveness and robustness of our proposed community-aware strategies. Moreover, the improvements made in the enhanced model are significant and consistent.

In this paper, our major contributions are as follows: 1) the investigation of the smoothing method using community context instead of the whole collection to enhance the language model for the document-based model; 2) the introduction of the community-sensitive AuthorRank for determining the query-sensitive authorities for experts, which is better than the query-independent version according to our experiment; and 3) the design of an adaptive ranking refinement strategy to aggregate the ranking results of both document-based model and community-sensitive AuthorRank, which leads to a significant improvement over the baseline method.

The rest of this paper is organized as follows. We briefly review some related work in Section II. In Section III, we present the expertise modeling based on the language model. In Section IV, we describe the enhanced models with community-aware authorities. In Section V, we define the experimental setup and report the experimental results. Finally, we present the conclusions and future work in Section VI.

II. Related Work

Since the introduction of an expert finding task in TREC 2005 [5], a great deal of work has been done in this area. Generally, there are two principal approaches for modeling expertise: candidate-based model and document-based model. These two models have been proposed and compared by Balog *et al.* in [6]. The candidate-based approach was first proposed by Craswell *et al.* [10], which is also referred to as profile-based method or query-independent approach in [6], [11]–[14]. In these methods, a profile ("virtual document") was built for each candidate based on all documents associated with the candidate, and the ranking scores were estimated according to the candidate profile in response to a given query. On the other hand, document-based models [6], [7], [13] are also referred to as query-dependent method in [14]. They first rank documents in the corpus for a given query topic and then find the associated candidates according to the retrieved documents. In terms of data management, candidate-based methods may require significantly smaller data in size than the original corpus. However, the contribution of each document in a profile cannot be measured individually. Meanwhile, the document-based model allows the application of advanced text modeling techniques in ranking individual documents, which achieves better performance than the candidate-based model. We choose the document-based model (i.e., Model 2 in [6]) as our baseline and propose several methods to further enhance this model.

Aside from the categories described previously, there are various methods proposed to extend or enhance expertise retrieval in many ways. Macdonald and Ounis presented a voting model for expert search in [15]. Their algorithm investigated several data fusion techniques to aggregate document votes into a ranking of candidates without using community-aware strategies. In [16], the authors extended the expert search by identifying some high-quality evidence. Bogers *et al.* [17] presented some relevant expert finding techniques that combine multiple sources of expertise evidence such as academic papers and social citation network. They only conducted experiments on a small-scale data collection, which makes their algorithms difficult to generalize to large-scale data collection. In [18], the authors proposed a graph-based reranking model and applied it to expert finding to refine ranking results. Serdyukov *et al.* [19] modeled the process of expert finding as multistep relevance propagation over the expertise graphs. Recently, Jiao *et al.* [20] proposed an ExpertRank algorithm for expert finding from online communities, while they combined the authority and relevance scores empirically without an adaptive refining strategy. Compared with previous methods, our proposed

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

community-aware strategies are different. In this paper, we utilize the AuthorRank [21] to measure the authority based on the coauthorship network [22], but it is query independent. We develop the query-sensitive AuthorRank as well as the adaptive ranking refinement strategy for the enhanced model.

More recently, several expert finding approaches used organization structure to help find experts. In [3], the authors employed the contextual information of organizational units to smooth the relevance scores. If an organization unit can be clustered as a community with similar research topics, the performance could be improved. Furthermore, Karimzadehgan *et al.* [23] leveraged the organizational hierarchy to enhance expert finding, in which they proposed a hierarchy-based algorithm to smooth the relevance scores using their neighbors' scores. They demonstrated that using the organizational hierarchy to propagate expertise scores can improve the effectiveness of expert finding algorithms. One deficiency of these algorithms is that it is quite difficult to obtain the organizational hierarchy in reality. Another problem is that the people in a same organization, for example, a computer science department, may be divided into totally different research groups, like hardware design community and data mining community. Thus, for a real-world academic domain, we consider the community instead of the organization to enhance expert finding.

Since we use community context information to smooth the language model, our work is related to existing works in statistical language modeling [24]–[27], which is employed to discover documents related to a query in the document-based model [12]. Ponte and Croft [24] were the first to apply the language modeling techniques in IR. From then on, many variations on these traditional language models have been developed to improve the performance of IR, such as relevance-based language model [28], title-based language model [29], and cluster-based language model [30]. Typically, a necessary and important step for the language model is to perform smoothing for the unseen query terms in the document, and several different smoothing methods have been proposed, such as Jelinek–Mercer smoothing and Bayesian smoothing using Dirichlet priors [26], [27]. However, all these smoothing methods only consider the collection as a whole, while our proposed smoothing method uses the community context information to smooth the language model instead of the whole collection.

Most of the previous work has been concentrated on expertise retrieval in enterprise corpora [6] or intranet data set [3]. By contrast, expert finding in academia domain has not been addressed much in the past. Li *et al.* [31] built an academic expertise-oriented search service, and they proposed a relevancy propagation-based algorithm [32] using the coauthorship network for expert finding. In addition, the expert finding task in a real-world academic field based on the DBLP bibliography was explored in [7], [33]. In this paper, we also focus on expert finding in an academic environment based on DBLP bibliography data.

III. Modeling Expertise Retrieval

The task of expertise retrieval is to retrieve a list of experts that have expertise for the given query. In this section, we first introduce the preliminaries and the baseline model for expertise retrieval and then describe statistical language modeling along with the new smoothing method using community context.

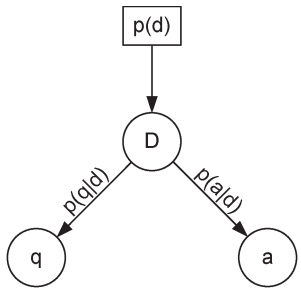

Fig. 2. Document-based model for expertise retrieval.

A. Preliminaries

Formally, suppose $A = \{a_1, a_2, \ldots, a_M\}$ is the set of expert candidates (i.e., authors) to be retrieved. Let $D = \{d_1, d_2, \ldots, d_N\}$ denote a collection of supporting documents, where d_i is a paper authored by one or several candidates. Let $\mathbb{C} = \{C_1, C_2, \ldots, C_K\}$ denote the collection of corresponding communities, where C_k consists of a set of papers and their associated authors. As shown in Fig. 1, the relationships between authors, documents, and communities can be represented by the tuple $\langle a_i, d_j, C_k \rangle$, signifying that author i has a paper j that is published in community k. Note that each paper exclusively belongs to one community, while an author may belong to multiple communities.

For a given query q, the problem of identifying experts is formulated using a generative probabilistic model, i.e., what is the probability of a candidate a_i being an expert given the query topic q? Using Bayes' theorem, the probability can be formulated as follows:

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

where $p(a_i, q)$ is the joint probability of a candidate and a query and $p(q)$ is the probability of the query. The probability $p(q)$ is a constant, so it can be ignored for ranking purposes. To derive the probability $p(a_i|q)$, it is equivalent to estimate the joint probability $p(a_i, q)$. In the rest of this section, we describe the detailed models on how to estimate this joint probability.

B. Document-Based Model

A number of methods have been proposed to estimate the probability $p(a_i, q)$. One successful document-based model, proposed by Deng *et al.* [7], decomposes the joint probability into the product over the supporting documents using a generative probabilistic model. The basic idea is to consider the expertise of an expert based on the relevance and importance of the associated documents. As shown in Fig. 2, the supporting documents D act as a "bridge" to connect the query q with candidate a. We follow this approach (document-based model) to estimate the probability as

$$\begin{aligned} p_d(a_i, q) &= \sum_{d_j \in D} p(d_j) p(a_i, q|d_j) \\ &= \sum_{d_j \in D} p(d_j) p(q|d_j) p(a_i|d_j, q) \end{aligned} \tag{2}$$

where $p(d_j)$ is the prior probability of a document, $p(q|d_j)$ means the relevance between q and d_j, and $p(a_i|d_j, q)$

TABLE I
COMBINATION OF DIFFERENT METHODS

Model	w[a]	c[b]	E[c]	Remarks
Document-based models				
$DM(b)$	B1	0	0	baseline model (i.e., Balog's Model 2 in [6])
$DM(bc)$	B1	1	0	build on top of $DM(b)$ with community-based smoothing
$DM(w)$	B2	0	0	weighted model with document priors (B2)
$DM(wc)$	B2	1	0	build on top of $DM(w)$ with community-based smoothing
Enhanced models				
$EDM(b)$	B1	0	1	enhancing $DM(b)$ with community-aware authorities
$EDM(bc)$	B1	1	1	enhancing $DM(bc)$ with community-aware authorities
$EDM(w)$	B2	0	1	enhancing $DM(w)$ with community-aware authorities
$EDM(wc)$	B2	1	1	enhancing $DM(wc)$ with community-aware authorities

[a] uniform weight (B1) or common logarithm weight (B2)
[b] smoothing using the community (1) or collection (0)
[c] enhancing with community-aware authorities (1) or no enhancement (0)

represents the association between the candidates and the documents for a given query. In this equation, we assume that candidate a is conditionally independent of the query q given a document d, i.e., $p(a_i|d_j, q) = p(a_i|d_j)$. Then, the aforementioned probability becomes

$$p_d(a_i, q) = \sum_{d_j \in D} p(d_j)p(q|\theta_{d_j})p(a_i|d_j) \tag{3}$$

where $p(q|\theta_{d_j})$ is the relevance score estimated using statistical language modeling.

Generally, the document prior $p(d)$ is assumed to be uniform, which leads to the document model proposed by Balog *et al.* in [6]. In addition, $p(d)$ can be interpreted as the document importance in [7], which is estimated based on the citation count of the document. These different settings of the document prior $p(d)$ could form two different models, as shown in Table I. We briefly define these two weight methods as follows:

$$p(d) \propto \begin{cases} 1, & (B1) \\ \log\left(10 + N_c(d)\right), & (B2) \end{cases} \tag{4}$$

where $N_c(d)$ is the citation count of d and the constant 10 is used to guarantee the weight to be greater than 1. The probability $p(a|d)$ indicates the association between papers and authors. One simple way is to define the probability inversely according to the number of authors. Suppose a document has multiple authors, each author is assumed to have the same knowledge about the topics described in the document

$$p(a|d) = \begin{cases} \frac{1}{N_a(d)}, & a \text{ is the author of } d \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $N_a(d)$ is the number of authors for the document.

C. Statistical Language Model

In the document-based model, one of the key challenges is to compute the relevance between a query and documents. In recent years, statistical language modeling has been widely used in the application of IR [24], [26], [27], [29]. To determine the probability of a query given a document, we infer a document language model θ_d for each document. The relevance score of document d with respect to query q is then defined as the conditional probability $p(q|\theta_d)$. Suppose $q = t_1, \ldots, t_m$ and each word t is generated independently, the relevance score would be

$$p(q|\theta_d) = \prod_{t_i \in q} p(t_i|\theta_d) \tag{6}$$

where $p(t|\theta_d)$ represents the maximum likelihood estimator of the word in a document d.

With such a model, the retrieval problem is reduced to the problem of estimating $p(t_i|\theta_d)$. In order to assign nonzero probabilities to unseen words, it is important to incorporate the smoothing methods in estimating the document language model. One popular way to smooth the maximum likelihood estimator is the Jelinek–Mercer smoothing method with the collection language model

$$p(t|\theta_d) = (1 - \lambda)\frac{n(t, d)}{|d|} + \lambda p(t|G) \tag{7}$$

where λ is the parameter to control the amount of smoothing, $n(t, d)$ is the count of word t in document d, $|d|$ is the number of words in d, and $p(t|G)$ is the collection language model. We follow Balog *et al.* [6] in setting $\lambda = 0.5$. Accordingly, we can define the collection-smoothed language model by substituting (7) into (6) as

$$p(q|\theta_d) = \prod_{t_i \in q} \left((1 - \lambda)\frac{n(t_i, d)}{|d|} + \lambda p(t_i|G) \right) \tag{8}$$

where the collection language model $p(t|G)$ can be estimated by normalizing the count of words in the entire collection as

$$p(t|G) = \frac{\sum_{d_j \in G} n(t, d_j)}{\sum_{d_j \in G} |d_j|}. \tag{9}$$

D. Smoothing Using Community Context

Now, we investigate how to use community information to enhance the language model described previously. In this section, a novel smoothing method is proposed for the document language model by leveraging the community-aware context to determine the probability $p(q|\theta_d)$.

Suppose the community information already exists for each document. For example, a conference or journal, which contains a set of publications, can be treated as a community.

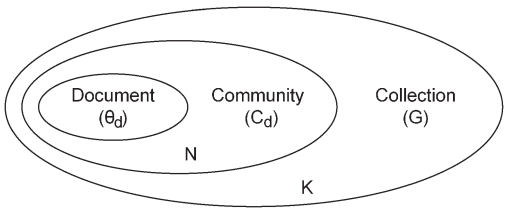

Fig. 3. Graph representation of the relationships between documents, communities, and the entire collection.

Fig. 3 shows the relationships between the documents, the communities, and the whole collection. There are three-level representations for the language model: the variable θ_d denotes the low-level document representation, sampled once per document; the variable C_d denotes the middle-level community representation, consisting of a set of documents including d; finally, the variable G denotes the high-level collection representation, consisting of all the documents.

According to the traditional language model, each word is smoothed by the same collection language model, which would be treated equally despite of their different community information. However, the community provides valuable insight and distinctive information for its documents because a document will somewhat share much more common information with its community rather than with the whole collection. Moreover, each community may have its own distinctive characteristics, which are different from other communities. Therefore, it would be more reasonable to employ the distinctive community language model, instead of the whole collection-based smoothing, to smooth different document models. The community language model is defined as

$$p(t|C_d) = \frac{\sum_{d_j \in C_d} n(t, d_j)}{\sum_{d_j \in C_d} |d_j|} \tag{10}$$

where C_d is the community that d_j belongs to. For two documents that belong to two different communities, we can define two distinctive community language models, instead of the same collection language model, to smooth the document language model. The community-smoothed language model is obtained by substituting $p(t|C_d)$ for $p(t|G)$ into (8) to obtain

$$\hat{p}(q|\theta_d) = \prod_{t_i \in q} \left((1-\lambda) \frac{n(t_i, d)}{|d|} + \lambda p(t_i|C_d) \right). \tag{11}$$

Note here that document d belongs to community C_d.

E. Combination of Different Methods

We have described two language models, i.e., as (8) and (11), for calculating $p(q|\theta_{d_j})$. Moreover, there are two methods in (4) for computing $p(d)$ as well. By setting $p(d)$ to be uniform as $B1$ in (4) and substituting (8) for $p(q|\theta_d)$ into (3), we obtain

$$p_d(a_i, q) = \sum_{d_j \in D} \left\{ \prod_{t_i \in q} \left((1-\lambda) \frac{n(t_i, d)}{|d|} + \lambda p(t_i|G) \right) \right\} p(a_i|d_j) \tag{12}$$

which is exactly the same as the document model (i.e., Model 2 on the "candidate centric" model) developed by Balog *et al.* in [6]. When considering each method and substituting into (3) separately, four different models can be combined, as shown in the upper part of Table I. Notice that Balog's Model 2, which is the best performing model in [6], is a special case of our model (denoted by $Balog_{m2}$). $DM(w)$ is a weighted model with document priors derived from citation count [i.e., $B2$ in (4)]. In addition, $DM(bc)$ and $DM(wc)$ are built on top of $DM(b)$ and $DM(w)$ with community-based smoothing method, respectively. To investigate the effect of the new developed smoothing method, we choose $Balog_{m2}$ as our baseline and compare the performance of other document-based models in Section V-C1.

IV. Enhanced Models With Community-Aware Authorities

In the academic domain, researchers in similar fields are most likely to form a community and to publish relevant articles in the community. Motivated by the observation that experts usually have high authority in some communities, we develop and investigate the query-sensitive authorities with an adaptive ranking refinement strategy, so as to enhance expertise retrieval models.

A. Discovering Authorities in a Community

In a community, the authors' relationships can be described using a coauthorship network, which has been used extensively to determine the structure of scientific collaborations [22]. We consider a weighted directed graph to model the coauthorship network in which each edge represents a coauthorship relationship. If any two authors have published a paper together, an edge with a weight is created. Let us take community C_1 in Fig. 1 as an example. Authors a_1 and a_2 coauthored paper d_1, and a_1, a_2, and a_3 coauthored paper d_2. Thus, a_1, a_2, and a_3 would be connected with each other.

To quantify the edge weight, the coauthorship frequency is proposed in [21], which consists of the sum of all values for all papers coauthored by a_i and a_j

$$f_{ij} = \sum_{k=1}^{N} \frac{\delta_i^k \delta_j^k}{n_k - 1} \tag{13}$$

where $\delta_i^k = 1$ if a_i is one of the authors of paper d_k; otherwise, $\delta_i^k = 0$ and n_k is the number of authors in paper d_k. This gives more weight to authors who copublish more papers together. For the aforementioned example, the graph with the coauthorship frequency is shown in Fig. 4(a). In general, the link weight w_{ij} from a_i to a_j is defined by normalizing the coauthorship frequency from a_i as

$$w_{ij} = \frac{f_{ij}}{\sum_{k=1}^{n} f_{ik}}. \tag{14}$$

This normalization ensures that the weights of an author's relationships sum to one, as shown in Fig. 4(b) for C_1.

For each community, a weighted coauthorship graph can be easily built. Intuitively, the generated coauthorship weights express valuable information which should, and can, be taken into account for discovering the authorities of the authors within the community. The PageRank algorithm [34] can be applied to a undirectional coauthorship graph by transforming each undirectional edge into a set of two directional symmetrical

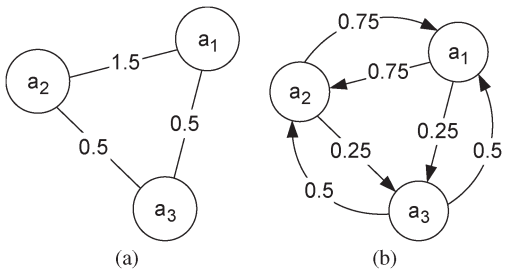

Fig. 4. Coauthorship graph with (a) coauthorship frequency and (b) normalized weight.

edges. Indeed, PageRank assumes that when a node a connects to n other nodes, each equally receives a fraction $1/n$ of $PR(a)$. However, in reality, the link weights express how strongly related two nodes, or authors, are in the coauthorship graph, and these weights can be used to determine the impact of an individual author in the network. We therefore utilize AuthorRank [21], a modification of PageRank [34] which considers link weight, to measure the authorities for the authors within this community. The AuthorRank of an author a_i is then given as follows:

$$p(a_i|C_k) = (1-\alpha)\frac{1}{N_a(C_k)} + \alpha \sum_{j=1}^{N_a(C_k)} w_{ij} \cdot p(a_j|C_k) \quad (15)$$

where $N_a(C_k)$ is the number of authors in the community C_k and $p(a_i|C_k)$ is the authority (i.e., AuthorRank) of the author a_i, satisfying $\sum_i p(a_i|C_k) = 1$. The AuthorRank can be calculated with the same iterative algorithm used by PageRank.

B. Community-Sensitive AuthorRank

The AuthorRank described earlier calculates the authority for the authors within a community, but it is independent of any particular query topic. To identify a set of experts for a given query, we propose a community-sensitive AuthorRank to generate query-specific authority scores for authors at query time. We precompute the authority scores offline for each community, as with ordinary AuthorRank. At query time, these authority scores are combined based on the communities of the query to form a composite AuthorRank score for those associated authors. Given a query q, we compute the probability for each community C_k the following:

$$p(C_k|q) = \frac{p(C_k) \cdot p(q|C_k)}{p(q)} \propto p(C_k) \prod_{t_i \in q} p(t_i|C_k) \quad (16)$$

where $p(t_i|C_k)$ is easily computed from the community language model as (10). The quantity $p(C_k)$ is not as straightforward. We model it in terms of the number of authors $N_a(C_k)$ and the average citation count per paper $\overline{N_c}(C_k)$ in the community C_k as

$$p(C_k) \propto N_a(C_k) \cdot \log\left(10 + \overline{N_c}(C_k)\right). \quad (17)$$

The number of authors reflects the size of the community, and the average citation count per paper reflects the quality of the community. Therefore, the underlying idea is that the community prior is proportional to the size and quality of the community.

According to (16), we retrieve the top k communities that are highly related to the query. Finally, we compute the query-sensitive authority score for each author as follows:

$$p(a_i|q) = \sum_k p(C_k|q)p(a_i|C_k). \quad (18)$$

The aforementioned community-sensitive AuthorRank has the following probabilistic interpretation. Note that (18) can be reformulated as

$$p(a_i|q) \propto p_c(a_i, q) = \sum_k p(C_k)p(q|C_k)p(a_i|C_k). \quad (19)$$

The authors are ranked according to this composite score $p_c(a_i, q)$. Supposing C_k is a "virtual" document, it becomes the document-based model (3). Thus, the community-sensitive AuthorRank can be regarded as a high-level document-based model that captures the high-level and general aspects for a given query.

C. Ranking Refinement Strategy

Based on the document-based model and the community-sensitive AuthorRank (i.e., community-based model), we obtain two kinds of ranking results $\vec{Rd}$ and $\vec{Rc}$, which reflect the authors' expertise from different perspectives. The ranking list $\vec{Rd}$ captures more specific and detailed aspects matching with the given query, as it measures the contribution of each document individually. In contrast, the ranking list $\vec{Rc}$ reflects more general and abstract aspects matching with the given query. In other words, if the document-based model is good for capturing the low-level and specific queries, then the community-sensitive AuthorRank should be good for capturing the high-level and general queries. Therefore, we consider the ranking refinement strategy by leveraging the community-sensitive AuthorRank to boost the document-based model.

In order to measure the similarity and diversity between two ranking results, we utilize a measurement, similar to the Jaccard coefficient, which is defined as the size of the intersection divided by the size of the union of these two top k ranking results

$$J = \frac{|\vec{Rd} \bigcap \vec{Rc}|}{|\vec{Rd} \bigcup \vec{Rc}|}. \quad (20)$$

This measurement implies the following meanings: A large value is reached if the community-sensitive AuthorRank could retrieve many common authors within the top k results as identified by the document-based model. In this case, the community-sensitive AuthorRank may contribute significantly to refine the document-based model; otherwise, vice versa. Based on this scheme, we adopt this measurement for an adaptive ranking refinement as follows. Let $Rd(a_i)$ be the rank of author a_i in $\vec{Rd}$. Suppose $\hat{Rc}$ is the subset of $\vec{Rc}$ consisting of the intersected authors $(\vec{Rd} \bigcap \vec{Rc})$, and let $\hat{Rc}(a_i)$ be the rank of author a_i in $\hat{Rc}$. For each author a_i in $\vec{Rd}$, we define a refined score $S(a_i)$ based on the following function:

$$S(a_i) = \frac{1}{Rd(a_i)} + \delta(a_i) \cdot J \cdot \frac{1}{\hat{Rc}(a_i)} \quad (21)$$

where $\delta(a_i) = 1$ if a_i is one of the intersected authors; otherwise, $\delta(a_i) = 0$. The intuition behind this method is that the

authors, which are identified in both $\vec{Rd}$ and $\vec{Rc}$, should be boosted ahead based on the ranking results $\vec{Rd}$. The new results are ranked according to the refined score $S(a_i)$. By applying the ranking refinement strategy to the previous four different document-based models, we obtain four enhanced models, as shown in Table I. The performances of these enhanced models are evaluated and compared in Section V-C2.

D. Overall Algorithm and Other Hybrid Methods

By unifying the document-based model in Section III and the enhanced model described earlier, we summarize the proposed algorithm in Algorithm 1. In the algorithm, note that we first perform preprocessing in a collection and precompute the following probabilities: $p(d_j)$, $p(a_i|d_j)$, $p(C_k)$ and $p(a_i|C_k)$. At query time, our approach is performed as shown in Algorithm 1. The document-based model is approximately performed using the top k_1 relevant documents obtained by top k algorithms [35], and meanwhile, the community-sensitive AuthorRank is implemented using the top k_2 relevant communities as well. In Section V-C4, we investigate and discuss the effect of these two parameters k_1 and k_2. In this paper, all of the algorithms used are programmed in C# language. We have implemented the language modeling approach to obtain the initial relevance scores with the Lucene.Net[2] package. For these experiments, the system indexes the collection and does tokenization, stopping, and stemming in the usual way.

Algorithm 1 Enhanced Expertise Retrieval Algorithm

Input: Given a query q,
Perform:

1) Retrieve the top k_1 most relevant documents based on the language model with (8) or (11);
2) Aggregate the expertise $p(a_i, q)$ using the document-based model (3) and then obtain the ranking results $\vec{Rd}$;
3) Identify the top k_2 most relevant communities according to (16);
4) Compute the community-sensitive AuthorRank with (18) and then obtain the ranking results $\vec{Rc}$;
5) Refine with (21) and get the new ranking results.

Output: Return the ranked experts $\{a_1, a_2, \ldots, a_k\}$.

To investigate the performance of the proposed refinement strategy, we compare the results using other two heuristic methods to combine the document-based model and the community-sensitive AuthorRank. One is a simple hybrid model by linear combination of the document-based model with the community-sensitive AuthorRank as

$$p_h(a_i, q) = \mu_1 p'_d(a_i, q) + (1 - \mu_1) p'_c(a_i, q) \tag{22}$$

where $p'_d(a_i, q)$ and $p'_c(a_i, q)$ are the normalized values of $p_d(a_i, q)$ and $p_c(a_i, q)$, respectively. Another one is a weighted

[2]http://incubator.apache.org/lucene.net/

```
<article mdate="2003-11-24" key="journals/cj/Fuhr92">
    <author>Norbert Fuhr</author>
    <title>Probabilistic Models in Information Retrieval.</title>
    <pages>243-255</pages>
    <year>1992</year>
    <volume>35</volume>
    <journal>Comput. J.</journal>
    <number>3</number>
    <url>db/journals/cj/cj35.html#Fuhr92</url>
</article>
```

Fig. 5. Sample of the DBLP XML records.

version of the reciprocal rank data fusion technique [36], which can be defined as

$$S'(a_i) = \mu_2 \frac{1}{Rd(a_i)} + (1 - \mu_2) \frac{1}{Rc(a_i)} \tag{23}$$

where $Rd(a_i)$ and $Rc(a_i)$ are the ranks of author a_i in Rd and Rc, respectively. The new results are ranked according to the aggregated scores $p_h(a_i, q)$ and $S'(a_i)$. By substituting (22) and (23) for (21) in Algorithm 1 individually, we could validate the aforementioned two hybrid methods and present the experimental results in Section V-C5.

As mentioned in Section IV-B, we investigate the community-sensitive (i.e., query-dependent) AuthorRank. In order to show the effectiveness of query-dependent aspects, we introduce the query-independent AuthorRank over multiple communities in the following. The aggregated query-independent AuthorRank can be regarded as a special case of query-dependent AuthorRank when we assume the relevance score $p(q|C_k)$ to be uniformly distributed as (19). The aggregated query-independent AuthorRank is then defined as follows:

$$p(a_i) \propto \sum_k p(C_k) p(a_i|C_k)$$

which leads to a static and query-independent ranking list Rc_s. By substituting $\vec{Rc_s}$ for $\vec{Rc}$ in Algorithm 1, we could evaluate a query-independent version of the enhanced model. In Section V-C5, our experimental results, as shown in Table X, confirm that the query-dependent version performs better than the query-independent version.

V. Experimental Evaluation

We evaluate the performance of our proposed models with different settings through an empirical evaluation. In this section, we first introduce the experimental setup, including the data set and evaluation metrics, and then present the experimental results.

A. Data Set

The data set that we study is the DBLP bibliography data, which contains over 1 100 000 XML records as of March 2009. Each record represents a paper that is originally published in conferences, journals, books, etc. One of the XML records is shown in Fig. 5, and it consists of several elements, such as "author," "title," "journal/conference," etc. In total, we gather about 700 000 author names from DBLP XML records, each of which can be an expert candidate. As the DBLP records contain limited information to represent the papers, we use a method

TABLE II
STATISTICS OF THE DBLP COLLECTION

Property	#of entities
Number of papers	1,152,512
Number of authors	695,906
Number of communities	3,311

similar to the one described in [7] to extend the information using Google Scholar[3] as a data supplement. For each paper, we use the title as the query to search in Google Scholar and select the top ten returned records as the supplemental data for this paper. The metadata (HTML pages) crawled from Google Scholar is up to 30 GB. This process is done automatically by a crawler and a parser, and the citation count of the paper in Google Scholar is obtained at the same time. In addition, we collect the community information according to the journals and conferences, and the total number of valid communities is 3311. For each community, we regard all the paper titles as the community context and construct the community coauthorship network for the affiliated authors. In summary, the data collection for experiments includes 1 152 512 papers, 695 906 authors, and 3311 communities, as shown in Table II.

The evaluation of expert finding performance in such a large data collection is very challenging due to the scarcity of ground truth that can be examined publicly. Furthermore, it is impractical to obtain expert ratings for all authors. In order to measure the performance of our proposed methods, a benchmark data set with 20 query topics and expert lists is manually created, as shown in Table III. The top nine topics and expert lists in the left table were collected by Zhang *et al.* [32], which are available at http://keg.cs.tsinghua.edu.cn/project/PSN/dataset.html#new_expert_list. The remaining 11 topics and relevance judgments for the corresponding expert lists were created and evaluated by ten researchers and senior graduate students of CUHK. Specifically, these 11 topics were created by the ten assessors based on their own research topics (i.e., IR, machine learning, and database). For example, three database people generated four query topics related to their own research, including privacy preservation, skyline, sensor RFID data management, and stream. Similarly, four machine learning people created some of their familiar subtopics, such as semisupervised learning, reinforcement learning, and kernel methods. We tried to cover not only broad queries (e.g., machine learning, IR) but also specific queries (e.g., language model for IR) to see whether our methods could handle both of them effectively.

Following general relevance judgments, for each query, a list of relevant experts is collected through the method of pooled relevance judgments with human assessment efforts. The top-ranked/retrieved authors from the computer science bibliography search engines (such as CiteSeer,[4] Libra,[5] Rexa,[6] and ArnetMiner[7]) and the committees of the top conferences related to the query topic were taken to construct the pools which contained around 300 authors. Moreover, since the query topics were created by the assessors who had conducted research in the related field for several years, they were quite familiar with the experts in that research field and could make reliable relevance judgments and even nominate some missing experts. The assessments were carried out mainly in terms of the number of top conference/journal papers an expert candidate had published, the number of related publications for the given query, and what distinguished awards he/she had received. There are four grade scores (3, 2, 1, and 0) which were assigned to represent top expert, expert, marginal expert, and not expert, respectively. For example, "W. Bruce Croft," who published over 60 papers in the Special Interest Group on Information Retrieval Conference (i.e., a top conference in the IR area), was marked as top expert for query "information retrieval." Basically, each query and the corresponding expert list are judged by at least three (to five) assessors, and we intentionally obtained a small number of experts by marking around 20 top-ranked experts as top experts (although the number of experts could be quite large). Finally, the judgment scores (at levels 3 and 2) were averaged to construct the final ground truth with 20–50 experts for each query, as shown in Table III.

B. Evaluation Metrics

For the evaluation of the task, three different metrics are employed to measure the performance of our proposed models, including precision at rank n (P@n), mean average precision (AP) (MAP), bpref [5], [37], [38], and mean reciprocal rank (MRR). P@n measures the fraction of the top n retrieved results that are relevant experts for the given query, which is defined as

$$P@n = \frac{\#\text{ relevant experts in top } n \text{ results}}{n}.$$

R-precision (R-prec) is defined as the precision at rank R where R is the number of relevant candidates for the given query. AP emphasizes returning more relevant documents earlier. For a single query, AP is defined as the average of the P@n values for all relevant documents

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

where n is the rank, N is the number retrieved, and $\text{rel}(n)$ is a binary function indicating the relevance of a given rank. MAP is the mean value of the APs computed for all the queries. Aside from the measurement of precisions, Bpref is a good score function that evaluates the performance from a different view, i.e., the number of nonrelevant candidates. It is formulated as

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

where r is a relevant candidate and n is a member of the first R candidates judged nonrelevant as retrieved by the system. The reciprocal rank of a query is the inverse of the rank of the first relevant result $rank_i$. The MRR is the average of the reciprocal ranks of results for a samples of queries Q

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}.$$

In our experiments, we report the results of P@10, P@20, P@30, R-prec, bpref, MAP, and MRR.

[3]http://scholar.google.com/
[4]http://citeseer.ist.psu.edu/
[5]http://libra.msra.cn/
[6]http://rexa.info/
[7]http://www.arnetminer.org/

TABLE III
BENCHMARK DATA SET OF 20 QUERIES

Topic	#Expert	Topic	#Expert
Information Extraction	20	Information Retrieval	23
Intelligent Agents	29	Language Model For Information Retrieval	12
Machine Learning	42	Face Recognition	21
Natural Language Processing	41	Semi Supervised Learning	21
Planning	34	Reinforcement Learning	17
Semantic Web	45	Kernel Methods	21
Support Vector Machine	31	Privacy Preservation	17
Boosting	56	Skyline	12
Ontology Alignment	53	Sensor RFID data management	13
Probabilistic Relevance Model	13	Stream	16

TABLE IV
EVALUATION AND COMPARISON OF DIFFERENT DOCUMENT-BASED METHODS. THE PERCENTAGES OF RELATIVE IMPROVEMENTS ARE SHOWN IN THE LOWER PART. BEST SCORES ARE IN BOLDFACE

Method	P@10	P@20	P@30	R-prec	MAP	bpref	MRR
$DM(b)$	0.505	0.425	0.3517	0.4082	0.3553	0.3522	0.7758
$DM(bc)$	0.525	0.4375	0.36	0.4206	0.3718	0.3679	0.8042
$DM(w)$	0.655	0.4775	0.4017	0.4664	0.4564	0.4412	0.9071
$DM(wc)$	**0.655**	**0.4925**	**0.4083**	**0.4747**	**0.4687**	**0.4532**	**0.91**
Relative Improvements (%)							
$DM(bc)/DM(b)$	+3.96%	+2.94%	+2.37%	+3.05%	+4.62%*	+4.47%*	+3.65%
$DM(wc)/DM(w)$	0%	+3.14%	+1.66%	+1.79%	+2.71%*	+2.72%*	+0.32%
$DM(w)/DM(b)$	+29.7%*	+12.35%*	+14.22%*	+14.25%*	+28.43%*	+25.28%*	+16.93%
$DM(wc)/DM(b)$	+29.7%*	+15.88%*	+16.11%*	+16.3%*	+31.91%*	+28.69%*	+17.29%

* indicates the improvement is statistically significant ($p < 0.05$).
$DM(b)$ is the baseline model, $DM(bc)$ is the community-smoothed $DM(b)$, $DM(w)$ is the weighted model with document prior, and $DM(wc)$ is the community-smoothed $DM(w)$.

C. Experimental Results

We present the experiments in five parts. First of all, the experiments are performed to compare the document-based models with different settings. Then, we examine the performance of the enhanced models after the ranking refinement. Next, we compare our enhanced models with other baselines. After that, we discuss the effect of two parameters by the empirical studies and show some detailed and intermediate results. Finally, we evaluate and compare two other hybrid methods.

1) Comparison of Document-Based Models: To validate the effect of the community-based smoothing method, we evaluate and compare the four document-based methods which are listed in Table I, including the baseline model $DM(b)$, weighted model $DM(w)$ with document prior B2, and their community-smoothed models $DM(bc)$ and $DM(wc)$. The results of these four methods are shown in Table IV. The first part shows the absolute precisions of these methods, and the second part illustrates the percentages of relevant improvements.

According to the first part, it is obvious that the community-smoothed model $DM(wc)$ which is built on top of weighted model $DM(w)$ achieves the best performance among the document-based models in all the metrics, such as 0.4925 for P@20 and 0.4687 for MAP. When looking at the relative improvements, we can see that community-smoothed model $DM(bc)$ improves over baseline model $DM(b)$ in all metrics, such as 3.96% for P@10 and 4.62% for MAP. Similarly, community-smoothed model $DM(wc)$ improves over weighted model $DM(w)$ in most metrics besides P@10 [it is harder to be improved as $DM(w)$ has been improved a lot over $DM(b)$]. Moreover, according to the t-test of statistical significance, we found that the improvements of both $DM(bc)/DM(b)$ and $DM(wc)/DM(w)$ are statistically significant in MAP and bpref ($p < 5\%$), as shown in Table IV. This is because the smoothing method using community context can boost the performance of the language model so as to improve the document-based model for expertise retrieval. The comparisons of $DM(w)/DM(b)$ and $DM(wc)/DM(b)$ show that the weighted model $DM(w)$ and the community-smoothed model $DM(wc)$ greatly improve the baseline $DM(b)$, which confirms the importance to consider the document prior in the document-based model. The t-test results indicate that the improvements of both $DM(w)/DM(b)$ and $DM(wc)/DM(b)$ are statistically significant in all metrics except MRR. The aforementioned experimental results demonstrate the effectiveness of the community-based smoothing method.

2) Comparison of Enhanced Models: In this section, we consider the question whether our proposed enhanced method can boost the performance by incorporating the document-based model with the community-sensitive AuthorRank. In Table V, we present the results of four enhanced models. A quick scan of the table reveals that enhanced model $EDM(wc)$ always outperforms other methods for all the metrics. In this table, we can see, as expected, that our proposed enhanced models perform better than their corresponding document-based models.

TABLE V
EVALUATION AND COMPARISON OF DIFFERENT ENHANCED METHODS WITH COMMUNITY-AWARE AUTHORITIES. THE PERCENTAGES OF RELATIVE IMPROVEMENTS ARE SHOWN IN THE LOWER PART. BEST SCORES ARE IN BOLDFACE

Method	P@10	P@20	P@30	R-prec	MAP	bpref	MRR
$EDM(b)$: Enhanced $DM(b)$	0.55	0.4675	0.3967	0.4443	0.3858	0.3987	0.7758
$EDM(bc)$: Enhanced $DM(bc)$	0.555	0.475	0.4017	0.4556	0.3993	0.4093	0.8042
$EDM(w)$: Enhanced $DM(w)$	0.675	0.5225	0.435	0.5097	0.4823	0.4803	0.9071
$EDM(wc)$: Enhanced $DM(wc)$	**0.68**	**0.535**	**0.4367**	**0.5109**	**0.4906**	**0.4876**	**0.91**
Relative Improvements (%)							
$EDM(b)/DM(b)$	+8.91%*	+10%*	+12.8%*	+8.85%*	+8.56%*	+13.2%*	0%
$EDM(bc)/DM(bc)$	+5.71%	+8.57%*	+11.58%*	+8.31%*	+7.74%*	+11.25%*	0%
$EDM(w)/DM(w)$	+3.05%	+9.42%*	+ 8.3%*	+9.28%*	+5.68%*	+8.85%*	0%
$EDM(wc)/DM(wc)$	+3.82%	+8.63%*	+ 6.94%*	+7.62%*	+4.66%*	+7.59%*	0%

* indicates the improvement is statistically significant ($p < 0.05$).

TABLE VI
COMPARISON OF ENHANCED MODEL $EDM(wc)$ WITH THE BASELINE MODELS BY BALOG *et al.* [6]

Method	P@10	P@20	P@30	R-prec	MAP	bpref	MRR
Balog's Model1	0.025	0.018	0.021	0.018	0.006	0.008	0.048
Balog's Model2	0.575	0.46	0.375	0.439	0.3915	0.3815	0.9167
$DM(b)$	0.505	0.425	0.3517	0.4082	0.3553	0.3522	0.7758
$EDM(wc)$	0.68	0.535	0.4367	0.5109	0.4906	0.4876	0.91
$EDM(wc)$/Balog's Model2	+18.3%*	+16.3%*	+16.4%*	+16.4%*	+25.3%*	+27.8%*	-0.73%

* indicates the improvement is statistically significant ($p < 0.05$).

As for the MAP metric, we measure a precision of 0.4906 for the enhanced model $EDM(wc)$, which improves the document-based model $DM(wc)$ by 4.66%. Similar results are also shown in the comparisons of $EDM(b)/DM(b)$, $EDM(bc)/DM(bc)$, and $EDM(w)/DM(w)$, and their relative improvements are 8.56%, 7.74%, and 5.68% for MAP, respectively. In terms of the comparisons using other metrics, we observe similar substantial improvements in the enhanced models as well. Regarding MRR metric, enhanced models have little effect/impact on the first relevant result. By comparing the precisions P@10, P@20, and P@30, an interesting observation is seen, which is that the quantities of improvements in P@20 and P@30 are more significant than those in P@10. The t-test results indicate these four relative improvements are statistically significant in most of the metrics except MRR, as shown in Table V. All the experimental results demonstrate the effectiveness of the enhanced models, which could further boost the performance of document-based models. Moreover, the improvements made in the enhanced models are consistent and promising. Therefore, it is very essential and promising to consider the enhanced models for expertise retrieval.

3) Comparison With Other Baselines: In this section, we compare our proposed models with other baseline models proposed by Balog *et al.* [6], including Model1 and Model2. To make such comparison fairly, we employ the source codes "EARS" from http://code.google.com/p/ears/ for Balog's models. Since our large-scale data set contains more than 1 million papers and about 0.7 million authors (i.e., expert candidates), we revised the original code appropriately so as to fit our data set. Table VI shows the evaluation results of our enhanced model $EDM(wc)$ and Balog's models, i.e., Model1 and Model2. It is obvious that the document-based model (i.e., Balog's Model2) outperforms the candidate-based model (i.e., Balog's Model1). Clearly, the performance of Balog's Model1 is very poor. The reason is that some authors who published a paper that happens to exactly match the query could obtain very high value in Model1. Moreover, there are lots of such data existing in our large-scale data set, so that top-ranked authors are dominated by such kinds of nonexperts, while the rank of real experts will be lowered in this model. However, Model2 obtains good results by aggregating the relevant papers for expert candidates. Indeed, our baseline model $DM(b)$ is exactly the same as Balog's Model2, but with slightly different implementations and parameters which lead to different results. Regarding the relative improvements of $EDM(wc)$/Balog's Model2, we can see that our enhanced model $EDM(wc)$ outperforms Balog's Model2, and the improvements are statistically significant for all the metrics except MRR. The aforementioned experimental results indicate the effectiveness of our proposed enhanced models.

4) Discussion and Detailed Results: We have shown the effectiveness and improvement of our proposed document-based models and enhanced models. The parameters k_1 and k_2 used in the previous sections are set to 5000 and 10, respectively. As mentioned before, we only retrieve the top k_1 relevant documents for the document-based model and identify top k_2 relevant communities for the community-sensitive AuthorRank as well. To investigate the effect of these two parameters, we designed the following experiments.

To examine the effect of k_1, we choose the best document-based model $DM(wc)$ [i.e., the community-smoothed model based on weighted model $DM(w)$] and evaluate it with four different values (from 1000 to 10 000). The experimental results for different k_1's are shown in Fig. 6(a). We can see that the performance becomes better for greater k_1's used in the document-based model. We believe that the reason is that more

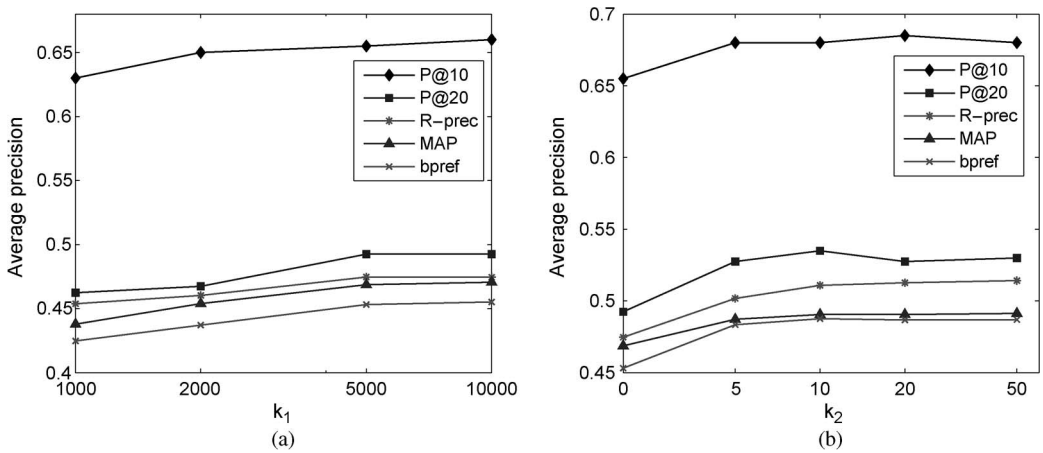

Fig. 6. Effect of varying the parameters (k_1 and k_2) in (a) the document-based model $DM(wc)$ and (b) the enhanced model $EDM(wc)$.

TABLE VII
DETAILED RESULTS OF THE COMMUNITY-SENSITIVE AUTHORRANK FOR THE QUERY "MACHINE LEARNING." THE FIRST ROW IS THE TOP FIVE COMMUNITIES FOR THE QUERY, AND THE REMAINING PART LISTS THE TOP-TEN-AUTHORS LISTS RANKED BY THEIR AUTHORITIES IN THE COMMUNITY

journals/ML	conf/ICML	conf/NIPS	journals/JMLR	conf/ECML
Pat Langley	Andrew W. Moore	Terrence J. Sejnowski	Michael I. Jordan	Saso Dzeroski
Robert E. Schapire	Sridhar Mahadevan	Michael I. Jordan	Yoram Singer	Johannes Frnkranz
Manfred K. Warmuth	Thomas G. Dietterich	Geoffrey E. Hinton	Tong Zhang	Gerhard Widmer
Thomas G. Dietterich	Prasad Tadepalli	Peter Dayan	Francis R. Bach	Ivan Bratko
Yoram Singer	Michael L. Littman	Christof Koch	Olivier Bousquet	Enric Plaza
Ryszard S. Michalski	Pat Langley	Klaus-Robert Mller	Klaus-Robert Mller	Pavel Brazdil
Michael J. Pazzani	Andrew McCallum	Zoubin Ghahramani	Bernhard Schlkopf	Birgit Tausend
Dana Angluin	Thorsten Joachims	Michael Mozer	Andr Elisseeff	Stephen Muggleton
Avrim Blum	Satinder P. Singh	Bernhard Schlkopf	Koby Crammer	Floriana Esposito
Leo Breiman	Michael I. Jordan	Satinder P. Singh	Ingo Steinwart	Stan Matwin

documents can better capture the complete expertise. However, a larger k_1 may result in longer processing time. Therefore, a good tradeoff is to set $k_1 = 5000$. To investigate the effect of k_2, we fix $k_1 = 5000$ and choose to compare the enhanced model $EDM(wc)$ with several different values from 0 to 50. Here, $k_2 = 0$ in $EDM(wc)$ represents its document-based model $DM(wc)$. As shown in Fig. 6(b), when incorporating the community-sensitive AuthorRank in the enhanced model $(k_2 > 0)$, the performance is improved compared with the document-based model $(k_2 = 0)$. The precisions first increase and then level off as k_2 grows. In general, the enhanced model $EDM(wc)$ is relatively robust for different k_2 and achieves good results when $k_2 = 10$.

To gain a better insight into the proposed enhanced model, we choose the query "machine learning" as the case to detail the combination of the community-sensitive AuthorRank and the document-based model and to show the intermediate results as well. We first present the detailed results of the community-sensitive AuthorRank in Table VII. According to (16), the top five relevant communities to the query "machine learning" are identified in the first row of Table VII, which are the "Machine Leaning journal," "ICML conference," "NIPS conference," "JMLR journal," and "ECML conference." Using AuthorRank, we could easily obtain their authorities for these communities. The top-ten-authors lists ranked by their authorities are listed in Table VII. As we can see, the proposed method can capture the right communities as well as the authoritative authors, such as "Andrew W. Moore" in ICML and "Michael I. Jordan" in NIPS. With the top k identified communities, the community-sensitive AuthorRank is employed to generate the query-sensitive authorities. In this case, the top-ten-authors list ranked by the query-sensitive authorities is shown in the second column of Table VIII. The other two columns in Table VIII report the top-ten-experts lists retrieved by $DM(wc)$ and $EDM(wc)$, respectively. We observe that a slight change occurs in the output of $EDM(wc)$ in contrast to that of $DM(wc)$, which would boost the persons retrieved by both the document-based model and the community-sensitive AuthorRank.

TABLE VIII
TOP-TEN-EXPERTS LISTS RETRIEVED BY THE DOCUMENT-BASED MODEL $DM(wc)$, THE COMMUNITY-SENSITIVE AUTHORRANK, AND THE ENHANCED MODEL $EDM(wc)$ FOR THE QUERY "MACHINE LEARNING"

DM(wc)	Authorities	EDM(wc)
Pat Langley	Pat Langley	Pat Langley
Thomas G. Dietterich	Robert E. Schapire	Thomas G. Dietterich
Sumio Watanabe	Manfred K. Warmuth	Sumio Watanabe
David E. Goldberg	Yoram Singer	David E. Goldberg
Tom M. Mitchell	Thomas G. Dietterich	Avrim Blum
Avrim Blum	Michael I. Jordan	Tom M. Mitchell
Ivan Bratko	Satinder P. Singh	Sanjay Jain
Donald Michie	Sanjay Jain	Ivan Bratko
Carl H. Smith	John Shawe-Taylor	Donald Michie
J. Ross Quinlan	Michael J. Pazzani	Michael I. Jordan

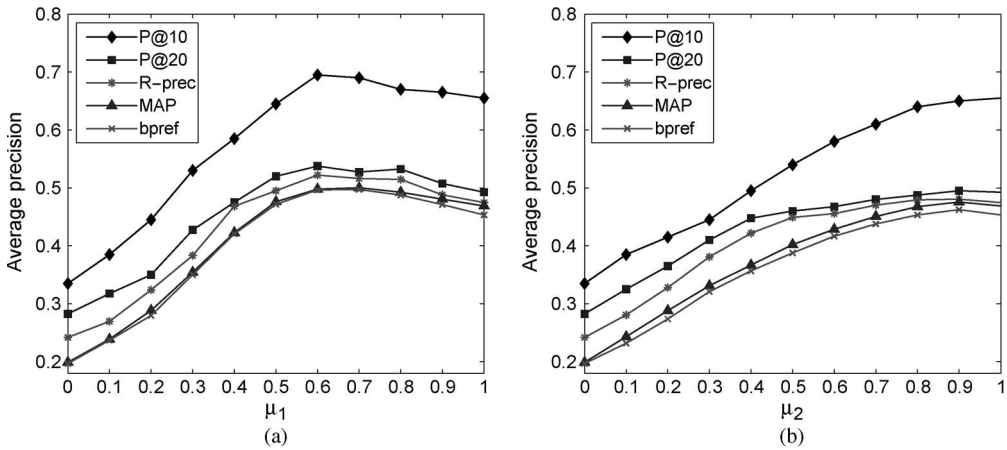

Fig. 7. Effect of the hybrid methods by tuning the parameters (μ_1 and μ_2) in (a) the hybrid method and (b) the reciprocal rank data fusion method. (a) Hybrid method ($p_h(a_i, q)$). (b) ($S'(a_i)$).

TABLE IX
EXPERIMENTAL RESULTS OF DIFFERENT HYBRID METHODS

Method	P@10	P@20	P@30	R-prec	MAP	bpref	MRR
$EDM(wc)$	0.68	0.535	0.4367	0.5109	0.4906	0.4876	0.91
Hybrid method ($\mu_1 = 0.6$)	0.695	0.5375	0.44	0.5221	0.4978	0.4965	0.91

TABLE X
COMPARISON OF ASPECTS OF QUERY-DEPENDENT AUTHORRANK AND QUERY-INDEPENDENT AUTHORRANK IN ENHANCED MODEL $EDM(wc)$

Method	P@10	P@20	P@30	R-prec	MAP	bpref	MRR
Query-independent (Q_i)	0.665	0.5	0.4133	0.4784	0.472	0.4554	0.91
Query-dependent (Q_d)	0.68	0.535	0.4367	0.5109	0.4906	0.4876	0.91
Q_d/Q_i Relative improvement	+2.26%	+7%*	+5.65%*	+6.79%*	+3.94%*	+7.07%*	0%

* indicates the improvement is statistically significant ($p < 0.05$).

5) Comparison of Different Hybrid Methods: In this section, we evaluate the other two hybrid methods by tuning μ from 0 to 1 with increments of 0.1. We restrict the combination of the best performing document-based model $DM(wc)$ and the community-sensitive AuthorRank. Fig. 7 shows the effect of varying the parameters on these methods. When $\mu = 1$, it is the document-based model $DM(wc)$; when $\mu = 0$, it is the community-sensitive AuthorRank. From Fig. 7(a), we notice that the hybrid method ($p_h(a_i, q)$) obtains the highest performance when $\mu_1 = 0.6$, which is better than both the document-based model $DM(wc)$ and the community-sensitive AuthorRank. However, the reciprocal rank data fusion method ($S'(a_i)$), as shown in Fig. 7(b), is always worse than the document-based model $DM(wc)$. The reason is that the ranking list obtained by the community-sensitive AuthorRank could provide valuable information as well as inaccurate information, which makes it hard to improve the performance without distinguishing the inaccurate and valuable information in the heuristic weighted version of the reciprocal rank data fusion method.

Our proposed ranking refinement strategy is similar to the reciprocal rank data fusion method. The difference is that we compute an adaptive value J to adjust the contribution of the ranking list obtained by the community-sensitive AuthorRank. Moreover, we distinguish the ranking list of the community-sensitive AuthorRank and choose the subset of the results that also appear in the ranking list of the document-based model as the valuable information. According to the valuable subset, the document-based model $DM(wc)$ could be further boosted to enhanced model $EDM(wc)$ by the proposed ranking refinement strategy. Table IX shows the results of enhanced model $EDM(wc)$ and the best performing hybrid method with parameter $\mu_1 = 0.6$. Comparing these two methods, $EDM(wc)$ can successfully achieve the best results of the hybrid method without the parameter setting, which shows the effect of the adaptive ranking refinement strategy.

The experiments described previously are executed with the query-dependent AuthorRank. Now, we conduct experiments to compare the aspects of query-dependent AuthorRank and query-independent AuthorRank. The evaluation results are shown in Table X, and the relative improvements of query-dependent version (Q_d) over query-independent version (Q_i) are statistically significant from P@20 to bpref, which confirms that the query-dependent version of the enhanced model outperforms the query-dependent version.

VI. CONCLUSION

In this paper, we have presented a set of community-aware strategies for enhancing expertise retrieval, including a new smoothing method based on community context and a community-sensitive AuthorRank based on coauthorship networks, which are motivated by the observation that the community provides valuable and distinctive information along with the documents and the experts. We not only formally

define and quantify these two strategies but also propose an adaptive ranking refinement method to incorporate both ranking results for an effective enhanced model. We apply the proposed models to the expert finding task on the DBLP bibliography data. Extensive experiments show that the improvements of our enhanced models are significant and consistent.

Although this work is a specialization of the expert retrieval task, it can be extended to other scenarios, e.g., to enterprise search where the community context can be modeled as similar documents by clustering, and the explicit coauthorship networks can be substituted with extracted co-occurrence networks. In future work, it would be interesting to apply the proposed methods to other entity retrieval problems with community information.

REFERENCES

[1] D. Petkova and W. B. Croft, "Proximity-based document representation for named entity retrieval," in *Proc. CIKM*, 2007, pp. 731–740.

[2] H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, and G. Attardi, "Ranking very many typed entities on wikipedia," in *Proc. CIKM*, 2007, pp. 1015–1018.

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

[4] Y. Cao, J. Liu, S. Bao, and H. Li, "Research on expert search at enterprise track of TREC 2005," in *Proc. TREC*, 2005. [Online]. Available: http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/c/Cao:Yunbo.html

[5] N. Craswell, I. Soboroff, and A. de Vries, "Overview of the TREC-2005 enterprise track," in *Proc. TREC*, 2005, pp. 199–205.

[6] K. Balog, L. Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise corpora," in *Proc. SIGIR*, 2006, pp. 43–50.

[7] H. Deng, I. King, and M. R. Lyu, "Formal models for expert finding on DBLP bibliography data," in *Proc. ICDM*, 2008, pp. 163–172.

[8] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha, "Probabilistic models for discovering e-communities," in *Proc. WWW*, 2006, pp. 173–182.

[9] H. Li, Z. Nie, W.-C. Lee, C. L. Giles, and J.-R. Wen, "Scalable community discovery on textual data with relations," in *Proc. CIKM*, 2008, pp. 1203–1212.

[10] N. Craswell, D. Hawking, A.-M. Vercoustre, and P. Wilkins, "Panoptic expert: Searching for experts not just for documents," in *Proc. Ausweb Poster*, Queensland, Australia, 2001.

[11] K. Balog and M. de Rijke, "Determining expert profiles (with an application to expert finding)," in *Proc. IJCAI*, 2007, pp. 2657–2662.

[12] K. Balog, L. Azzopardi, and M. de Rijke, "A language modeling framework for expert finding," *Inf. Process. Manage.*, vol. 45, no. 1, pp. 1–19, Jan. 2009.

[13] H. Fang and C. Zhai, "Probabilistic models for expert finding," in *Proc. ECIR*, 2007, pp. 418–430.

[14] D. Petkova and W. B. Croft, "Hierarchical language models for expert finding in enterprise corpora," in *Proc. ICTAI*, 2006, pp. 599–608.

[15] C. Macdonald and I. Ounis, "Voting for candidates: Adapting data fusion techniques for an expert search task," in *Proc. CIKM*, 2006, pp. 387–396.

[16] C. Macdonald, D. Hannah, and I. Ounis, "High quality expertise evidence for expert search," in *Proc. ECIR*, 2008, pp. 283–295.

[17] T. Bogers, K. Kox, and A. van den Bosch, "Using citation analysis for finding experts in workgroups," in *Proc. DIR*, 2008, pp. 21–28.

[18] H. Deng, M. R. Lyu, and I. King, "Effective latent space graph-based re-ranking model with global consistency," in *Proc. WSDM*, 2009, pp. 212–221.

[19] P. Serdyukov, H. Rode, and D. Hiemstra, "Modeling multi-step relevance propagation for expert finding," in *Proc. CIKM*, 2008, pp. 1133–1142.

[20] J. Jiao, J. Yan, H. Zhao, and W. Fan, "ExpertRank: An expert user ranking algorithm in online communities," in *Proc. NISS*, 2009, pp. 674–679.

[21] X. Liu, J. Bollen, M. L. Nelson, and H. V. de Sompel, "Co-authorship networks in the digital library research community," *Inf. Process. Manage.*, vol. 41, no. 6, pp. 1462–1480, Dec. 2005.

[22] M. Newman, "Coauthorship networks and patterns of scientific collaboration," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 101, no. suppl 1, pp. 5200–5205, Apr. 2004.

[23] M. Karimzadehgan, R. W. White, and M. Richardson, "Enhancing expert finding using organizational hierarchies," in *Proc. ECIR*, 2009, pp. 177–188.

[24] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proc. SIGIR*, 1998, pp. 275–281.

[25] J. D. Lafferty and C. Zhai, "Document language models, query models, and risk minimization for information retrieval," in *Proc. SIGIR*, 2001, pp. 111–119.

[26] C. Zhai, "Statistical language models for information retrieval: A critical review," *Found. Trends Inf. Retrieval*, vol. 2, no. 3, pp. 137–215, 2008.

[27] C. Zhai and J. D. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Trans. Inf. Syst.*, vol. 22, no. 2, pp. 179–214, Apr. 2004.

[28] V. Lavrenko and W. B. Croft, "Relevance-based language models," in *Proc. SIGIR*, 2001, pp. 120–127.

[29] R. Jin, A. G. Hauptmann, and C. Zhai, "Title language model for information retrieval," in *Proc. SIGIR*, 2002, pp. 42–48.

[30] X. Liu and W. B. Croft, "Cluster-based retrieval using language models," in *Proc. SIGIR*, 2004, pp. 186–193.

[31] J.-Z. Li, J. Tang, J. Zhang, Q. Luo, Y. Liu, and M. Hong, "EOS: Expertise oriented search using social networks," in *Proc. WWW*, 2007, pp. 1271–1272.

[32] J. Zhang, J. Tang, and J.-Z. Li, "Expert finding in a social network," in *Proc. DASFAA*, 2007, pp. 1066–1069.

[33] H. Deng, I. King, and M. R. Lyu, "Enhancing expertise retrieval using community-aware strategies," in *Proc. CIKM*, 2009, pp. 1733–1736.

[34] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comput. Netw.*, vol. 30, no. 1–7, pp. 107–117, Apr. 1998.

[35] R. Fagin, R. Kumar, and D. Sivakumar, "Comparing top k lists," in *Proc. SODA*, 2003, pp. 28–36.

[36] M. Zhang, R. Song, C. Lin, S. Ma, Z. Jang, Y. Lin, Y. Liu, and L. Zhao, "Expansion-based technologies in finding relevant and new information: THU TREC2002: Novelty track experiments," in *Proc. TREC*, 2002, p. 591.

[37] C. Buckley and E. M. Voorhees, "Retrieval evaluation with incomplete information," in *Proc. SIGIR*, 2004, pp. 25–32.

[38] I. Soboroff, A. de Vries, and N. Craswell, "Overview of the TREC-2006 enterprise track," in *Proc. TREC*, 2006. [Online]. Available: http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/s/Soboroff:Ian.html

Hongbo Deng received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Shatin, Hong Kong, in 2009.

He is currently a Research Scientist with the Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana. His research interests include information network analysis, data mining, Web search, and their wide applications in query log analysis, scientific literature, and other textual data. He has published a number of papers in the leading Web search and data mining conferences, and has recently served on the program committees of several leading conferences and workshops.

Irwin King (SM'08) received the B.Sc. degree in engineering and applied science from the California Institute of Technology, Pasadena, and the M.Sc. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles.

In 1993, he joined The Chinese University of Hong Kong, Shatin, Hong Kong, where he is currently a Professor with the Department of Computer Science and Engineering. He is currently on leave to be with AT&T Labs Research, San Francisco, CA, and is also a Visiting Professor at the School of Information, University of California, Berkeley. His research interests include machine learning, social computing, Web intelligence, data mining, and multimedia processing. In these research areas, he has over 210 technical publications in journals and top conferences. In addition, he has contributed over 20 book chapters and edited volumes. Moreover, he has over 30 research and applied grants. One notable system he has developed is the VeriGuide system, which detects similar sentences and performs readability analysis of text-based documents in both English and Chinese to promote academic integrity and honesty. He is the Book Series Editor for "Social Media and Social Computing" with Taylor and Francis (CRC Press). He has served as Special Issue Guest Editor for *Neurocomputing*, the *International Journal of Intelligent Computing and Cybernetics*, the *Journal of Intelligent Information Systems*, and the *International Journal of Computational Intelligent Research*.

Dr. King is a member of the Association for Computing Machinery, the International Neural Network Society (INNS), and the Asian Pacific Neural Network Assembly (APNNA). Currently, he is serving the Neural Network Technical Committee and the Data Mining Technical Committee under the IEEE Computational Intelligence Society (formerly the IEEE Neural Network Society). He is also an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE. He is a member of the Board of Governors of INNS and a Vice President and Governing Board Member of APNNA. He also serves INNS as the Vice President for Membership in the Board of Governors. In addition, he has served as a member in the Research Grants Council Engineering Panel for the Hong Kong SAR Government, in the Review Panel of the Natural Sciences and Engineering Research Council of Canada, and also in the Review Panel of the Research Council for Natural Sciences and Engineering of the Academy of Finland.

Michael R. Lyu (F'04) received the B.Sc. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1981, the M.Sc. degree in computer engineering from the University of California, Santa Barbara, in 1985, and the Ph.D. degree in computer science from the University of California, Los Angeles, in 1988.

He was with the Jet Propulsion Laboratory, Pasadena, CA, from 1988 to 1990; with the Department of Electrical and Computer Engineering, University of Iowa, Ames, from 1990 to 1992; with Bell Communications Research (Bellcore), Morristown, NJ, from 1992 to 1995; and with Bell Laboratories, Murray Hill, NJ, from 1995 to 1997. In 1998, he joined The Chinese University of Hong Kong, Shatin, Hong Kong, where he is currently a Professor with the Department of Computer Science and Engineering. He is also the Founder and Director of the Video over InternEt and Wireless (VIEW) Technologies Laboratory. His research interests include software reliability engineering, distributed systems, fault-tolerant computing, mobile and sensor networks, Web technologies, multimedia information processing and retrieval, and machine learning. He has published 300 refereed journal and conference papers in these areas. He was the Editor of two book volumes: *Software Fault Tolerance* (New York: Wiley, 1995) and *The Handbook of Software Reliability Engineering* (Piscataway, NJ: IEEE and New York: McGraw-Hill, 1996).

He initiated the First International Symposium on Software Reliability Engineering (ISSRE) in 1990. He was the Program Chair for ISSRE 1996 and General Chair for ISSRE 2001. He was also PRDC 1999 Program Cochair, WWW10 Program Cochair, SRDS 2005 Program Cochair, PRDC 2005 General Cochair, and ICEBE 2007 Program Cochair. He was on the editorial board of the *Journal of Information Science and Engineering* and the *Wiley Software Testing, Verification and Reliability Journal*.

Dr. Lyu is an American Association for the Advancement of Science fellow and a Croucher senior research fellow. He was on the editorial board of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING and the IEEE TRANSACTIONS ON RELIABILITY.