Co-occurrence Based Diffusion for Expert Search On the Web

Ziyu Guan, Gengxin Miao, Russell McLoughlin, Xifeng Yan, *Member, IEEE,*
Deng Cai, *Member, IEEE*

Abstract—Expert search has been studied in different contexts, e.g. enterprises, academic communities. We examine a general expert search problem: searching experts on the Web, where millions of Web pages and thousands of names are considered. It has mainly two challenging issues: (1) Web pages could be of varying quality and full of noises; (2) The expertise evidences scattered in Web pages are usually vague and ambiguous. We propose to leverage the large amount of co-occurrence information to assess the relevance and reputation of a person name for a query. The co-occurrence structure is modeled by a hypergraph, on which a heat diffusion based ranking algorithm is proposed. Query keywords are regarded as heat sources, and a person name which has strong connection with the query (i.e. frequently co-occur with query keywords and other names related to query keywords) will receive most of the heat, thus being ranked high. Experiments on the ClueWeb09 Web collection show that our algorithm is effective for retrieving experts and outperforms baseline algorithms significantly. This work would be regarded as one step towards addressing the more general entity search problem without sophisticated NLP techniques.

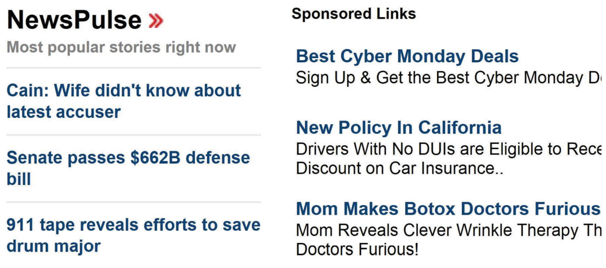
Index Terms—Expert search, web mining, co-occurrence, diffusion

1 Introduction

Expert search gained increasing attention from both industry and academia. The TREC enterprise tracks [16] boomed research work on organizational expert search [2], [8], [34], [43], [20]. Variant expert search problems were also identified and addressed in other domains such as question answering [26], online forums [41] and academic society [31], [19], [42].

However, previous work on expert search is often confined within specific contexts, e.g. an enterprise corpus, an online forum, or an academic bibliography collection. Recently, the desire to find experts on a variety of daily life topics is increasing. We are observing a rising search paradigm that allows users to search for people who can answer their natural language questions [22]. However, this system requires users to register and join a community. In contrast, the Web contains a huge amount of information about people (e.g. personal home pages, blogs, Web news). It is possible to build a powerful expert search engine by exploiting the information about people on the Web.

In this paper we propose a general expert search problem: expert search on the Web, which considers ordinary Web pages and people names. This problem is different from organizational expert search and is more like Google where our goal is to return a list of experts with reasonable quality. It has new challenges: (1) Compared to an organization's repository, ordinary Web pages could be of varying quality (e.g. spam [32]) and full of noises. Fig. 1 shows examples of noises from a news page of CNN, i.e. links to popular news stories and advertisements, which are usually irrelevant to the current story. (2) The expertise evidences scattered in Web pages are usually vague and ambiguous. Fig. 2 shows a snippet from the Wikipedia page of Ana Ivanovic, a former World No. 1 tennis player. However, one can find there are many Web pages saying she used to train in a swimming pool, though she is not an expert in swimming.



NewsPulse »
Most popular stories right now

Cain: Wife didn't know about latest accuser

Senate passes $662B defense bill

911 tape reveals efforts to save drum major

Sponsored Links

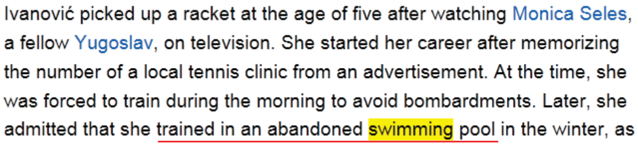
Best Cyber Monday Deals
Sign Up & Get the Best Cyber Monday D

New Policy In California
Drivers With No DUIs are Eligible to Rece
Discount on Car Insurance..

Mom Makes Botox Doctors Furious
Mom Reveals Clever Wrinkle Therapy Th
Doctors Furious!

(a) (b)

Fig. 1. Example noises in Web pages.



Ivanović picked up a racket at the age of five after watching Monica Seles, a fellow Yugoslav, on television. She started her career after memorizing the number of a local tennis clinic from an advertisement. At the time, she was forced to train during the morning to avoid bombardments. Later, she admitted that she trained in an abandoned swimming pool in the winter, as

Fig. 2. A vague expertise evidence.

- *Z. Guan, R. McLoughlin and X. Yan are with the Department of Computer Science, University of California, Santa Barbara, CA 93106. E-mail: ziyuguan@cs.ucsb.edu*
- *G. Miao is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106. E-mail: miao@umail.ucsb.edu*
- *D. Cai is with State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, CN 310027. E-mail: dengcai@cad.zju.edu.cn*

In traditional organizational expert search, relevance is the major concern. However, considering the challenges mentioned above, we also need to consider a name's reputation for a query topic as well as the trustworthiness of data sources. We suspect the relevance and reputation can be captured by the large amount of *keyword-name* and *name-name* co-occurrences on the Web. Using a large amount of co-occurrence information, noises could be suppressed since noisy co-occurrences would not appear frequently on the Web. The problem in Fig. 2 can thus be alleviated because Ana Ivanovic probably does not co-occur frequently with salient swimmers. In particular, we aim to address the new challenging issues by leveraging the linkage of experts exhibited on the Web: (1) **Relevance**. Related experts should co-occur frequently on many Web pages with the keywords in the query. (2) **Reputation**. Related experts should co-occur frequently with other people related to the query, regardless of whether they are experts or not. For example, a salient researcher could be co-mentioned with other researchers in his/her research areas many times; a senior user in an online forum would actively pursue threads for which he/she has expertise and co-occur with many other users. (3) **Trustworthiness**. Related experts tend to occur in high quality Web pages.

The second observation could be true for many domains, since humans are socialized and social activities shall be reflected on the Web. Following these observations, we propose to model the co-occurrence relationships among people names and words by a heterogeneous hypergraph where Web pages are treated as hyperedges with PageRank scores as their weights. Then we develop a novel heat diffusion model on the hypergraph. Based on this model, an expert ranking algorithm, called Co-occurrence Diffusion (CoDiffusion for short), is developed. Given a query, we treat keywords in the query as heat sources and perform heat diffusion. Names with the highest heat scores are returned. Intuitively, people who have strong connection with the query (i.e. frequently co-occur with query keywords and frequently co-occur with other people related to query keywords in high quality pages) will be ranked high.

Connection with Renlifang. Renlifang[1] is an object level search engine which allows users to query about people, locations, and organizations and explore their relationships. It extracts structural information about entities and their relationships by deep-parsing Web pages [36], [27], [44]. In contrast, CoDiffusion does not rely on complicated natural language processing techniques to search experts. While Renlifang does have an expert finding function, its ranking algorithm seems not publicly known.

Our contributions. A major contribution of this study is an examination of a new expert search problem: searching experts on the Web, and the proposal of utilizing co-occurrence relationships to assess the relevance and reputation of a person name with respect to a query simultaneously. This work would be regarded as one step towards addressing the more general entity search problem without sophisticated NLP techniques, where different types of entities are considered, e.g. people, organizations, locations. We abstract the co-occurrence relationships using a heterogeneous hypergraph and develop a novel heat diffusion method on this hypergraph to address the expert search problem. The diffusion method considers both relevance and reputation for ranking experts, as well as the quality of data sources. We also try to boost performance by re-ranking based on name pseudo relevance feedback. Empirical results on the ClueWeb09 Web collection[2] show that our method outperforms baseline methods and well-known language model-based approaches significantly. We also demonstrate the usefulness of people co-occurrence information in ranking experts. We are not going to discuss person name extraction and disambiguation [1], [39], [35], [11], which are out of scope of this work.

2 Related Work

Expert search is a growing research area. Early approaches for expert search involve building a knowledge base which contains the descriptions of people's skills within an organization [15]. However, creating a knowledge base manually is time-consuming and laborious. Therefore, automatic approaches have been developed for building people profiles [17], [38]. Expert search became a hot research area since the start of the TREC enterprise track [16] in 2005. A lot of studies were dedicated to organizational expert search. Balog *et al.* proposed a language model framework for expert search [2]. Their Model 1 is equivalent to a profile-centric approach where text from all the documents associated with a person is amassed to represent that person. Their Model 2 is a document-centric approach which first computes the relevance of documents to a query and then accumulates for each person the relevance scores of the documents that are associated with the person. This process was formulated in a generative probabilistic model. Balog *et al.* showed that Model 2 outperformed Model 1 [2] and it became one of the most prominent methods for expert search. In their following work, Balog *et al.* tried to apply and refine their language model on a smaller dataset comprising multilingual data crawled from Tilburg University's Website [4].

Researchers have investigated using additional information to boost retrieval performance, such as PageRank, indegree, and URL length of documents

[1] http://renlifang.msra.cn/

[2] http://boston.lti.cs.cmu.edu/Data/clueweb09/

[43], person-person similarity [4], internal document structures that indicate people's association with document content [6], query expansion and relevance feedback using people names [30], [7], non-local evidence [8], [33], proximity between occurrences of query words and people names [21], [3]. Besides language models, other methods have been proposed. Macdonald and Ounis proposed a method based on voting and data fusion techniques [29]. Serdyukov *et al.* modeled associations between people and documents as a bipartite graph and performed probabilistic random walks to find relevant experts [34]. Fang *et al.* proposed a relevance-based discriminative learning framework for expert search [20]. Many other methods for organizational expert search were proposed during TREC Enterprise tracks.

Two benchmark datasets, W3C [16] and CSIRO [9], are the focus of these organizational expert search works, which are crawls of the Websites of W3C and Commonwealth Scientific and Industrial Research Organization, respectively. However, searching experts on the Web is different from organizational expert search in that we consider ordinary Web pages and people names. Besides relevance, in our case we also need to consider the reputation of a name. This is because (1) compared to an organization's Website or document repository, Web collections could be of low quality and noisy; (2) the expertise information contained in ordinary Web pages could be vague. In this paper, we propose to use co-occurrences to assess the relevance and reputation of a person name with respect to a query simultaneously and we will demonstrate its effectiveness in experiments.

There are other expert retrieval problems. Balog and de Rijke studied the problem of finding similar experts, given example experts [5]. Zhang *et al.* studied characteristics of online forums and tested using link analysis methods to identify users with high expertise [41]. Liu *et al.* studied expert finding in community-based question answering Websites and treated it as an IR problem [26]. Mimno and McCallum used topic modeling to address the problem of matching papers with reviewers [31]. Later Karimzadehgan *et al.* addressed this review assignment problem based on matching of multiple aspects of expertise [24], [23]. Deng *et al.* explored using language modeling and a topic-based model for expert finding in the DBLP bibliography data [19]. Zhou *et al.* proposed co-ranking authors and their publications using coupled random walks [42].

Recently, the idea of heat diffusion was extended to the discrete graph setting, with applications such as dimension reduction [12], classification [25], topic modeling [14], matrix factorization [13], anti-spamming [37], social network marketing [28] and online advertisement matching [10]. These studies considered diffusion in homogeneous graphs. We develop a diffusion model on heterogeneous hypergraphs for our expert search problem.

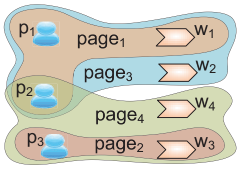


Fig. 3. An example heterogeneous hypergraph.

3 Heat Diffusion on Heterogeneous Hypergraphs

3.1 Notations and Problem Formulation

In a hypergraph, edges (called hyperedges) can connect two or more vertices. Formally, let $G = (V, E)$ be a hypergraph with vertex set V and edge set E. A hyperedge $e \in E$ can be regarded as a subset of vertices. e is said to be incident with a vertex v if $v \in e$. Each hyperedge e is associated with a weight denoted by $w(e)$. In our case, there are three types of objects: people (names), words, and Web pages, denoted by $\mathcal{P}$, $\mathcal{W}$ and $\mathcal{D}$, respectively. By the co-occurrence relationships among $\mathcal{P}$ and $\mathcal{W}$ established by Web pages, we can construct a heterogeneous hypergraph $G_{\mathcal{P},\mathcal{W}} = (V, E)$ where V contains all the people and words and each $e \in E$ corresponds to a Web page (Fig. 3). $w(e)$ is the PageRank score of e's corresponding Web page. The problem is, *given $\mathcal{P}$, $\mathcal{W}$, $G_{\mathcal{P},\mathcal{W}}$ and query keywords from $\mathcal{W}$, to rank $\mathcal{P}$ according to their expertise in the topic represented by the query.*

We propose using heat diffusion to address this ranking problem. Let V_p and V_w represent the vertex sets corresponding to people and words, respectively. Consequently, $V = V_p \cup V_w$. Let $\mathbf{H}_p$ be a $|V_p| \times |E|$ weighted incidence matrix where an entry $H_p(v, e) = wt_{v,e}$ if $v \in e$ ($v \in V_p$) and 0 otherwise. $\mathbf{H}_w$ is defined similarly for V_w. $wt_{v,e}$ reflects the connection strength between object v and page e. We set $H_p(v, e)$ to the number of times person v appears in page e and set $H_w(v, e)$ to the TF-IDF score of word v in e. The degree of a vertex v is defined as

$$d(v) = \left\{ \begin{array}{ll} \sum_{e \in E} w(e) H_p(v, e) & v \in V_p \\ \sum_{e \in E} w(e) H_w(v, e) & v \in V_w \end{array} \right. . \tag{1}$$

The degree of a hyperedge is defined as

$$\delta(e) = \delta_p(e) + \delta_w(e), \tag{2}$$

where $\delta_p(e) = \sum_{v \in V_p} H_p(v, e)$ and $\delta_w(e) = \sum_{v \in V_w} H_w(v, e)$. We define $f_i^p(t)$ and $f_j^w(t)$ to be the heat of vertex $i \in V_p$ and that of vertex $j \in V_w$ at time t, respectively. Let $\mathbf{f}^p(t)$ and $\mathbf{f}^w(t)$ be the heat distribution vectors at time t with sizes $|V_p| \times 1$ and $|V_w| \times 1$, respectively. The initial heat distribution is represented by $\mathbf{f}^p(0)$ and $\mathbf{f}^w(0)$. Then the problem is to derive the heat distribution at time t ($\mathbf{f}^p(t)$ and $\mathbf{f}^w(t)$) given an initial distribution at time 0 ($\mathbf{f}^p(0)$

and $\mathbf{f}^w(0)$). In other words, we can set query objects (people and/or words) as heat sources and rank other objects according to the heat distribution at time t, which reflects the affinity between the objects and heat sources. This is a general ranking model. In our problem, words are queries and we need to get the ranking of people.

3.2 Diffusion Model

In real world, heat diffuses in a medium from positions with higher temperatures to those with lower temperatures. The most important property of heat diffusion is that the heat flow rate at a point is proportional to the second order derivative of heat with respect to the space at that point [25], [37]:

$$\frac{\partial f(\mathbf{x},t)}{\partial t} = \gamma \nabla^2 f(\mathbf{x},t), \tag{3}$$

where $f(\mathbf{x},t)$ represents the heat at position $\mathbf{x}$ and at time t, ∇^2 denotes the Laplacian operator and γ is the thermal conductivity coefficient. In the following, we present our diffusion model for heterogeneous hypergraphs.

Different medium have different thermal conductivity coefficients. Therefore, we define three coefficients: γ_{pp}, γ_{ww} and γ_{pw} to characterize the thermal conductivity among people, among words and between people and words, respectively. The diffusion model is constructed as follows. At time t, each vertex $i \in V$ will receive an amount of heat from its neighbors (i.e. neighboring people and words) during a small time period Δt. In other words, regarding hyperedges as pipes connecting vertices, i will receive heat from all hyperedges which contain i. For each $i \in V_p$, the amount of heat it receives from j through hyperedge e which contains both i and j should be proportional to (1) the time period Δt, (2) the conductivity coefficients γ_{pp} (if $j \in V_p$) or γ_{pw} (if $j \in V_w$), (3) the edge weight $w(e)$: heat diffuses quickly through high quality Web pages, and (4) the heat difference $f_j^p(t) - f_i^p(t)$ (if $j \in V_p$) or $f_j^w(t) - f_i^p(t)$ (if $j \in V_w$). Therefore, the heat difference at vertex i between $t + \Delta t$ and t is

$$\begin{aligned} & f_i^p(t+\Delta t) - f_i^p(t) \\ & = \sum_{e\in E} H_p(i,e)\frac{w(e)}{\delta_p(e)} \sum_{j\in V_p} H_p(j,e)[\frac{f_j^p(t)}{d'(j)} - \frac{f_i^p(t)}{d'(i)}]\gamma_{pp}\Delta t \\ & + \sum_{e\in E} H_p(i,e)\frac{w(e)}{\delta_w(e)} \sum_{j\in V_w} H_w(j,e)[\frac{f_j^w(t)}{d'(j)} - \frac{f_i^p(t)}{d'(i)}]\gamma_{pw}\Delta t, \end{aligned} \tag{4}$$

where we take the sums over all $e \in E$ and $j \in V_p$ and $j \in V_w$ since unrelated objects have zero incidence values, i.e. $H_p(v,e)$ or $H_w(v,e)$. There are several normalization terms in the above equation. We use $\delta_p(e)/\delta_w(e)$ to normalize $w(e)$ in that if a Web page contains many people/words, then the connection between any of those people/words and person i in e should be weak. We use the connectivity (i.e. $d'(v)$) of a vertex to normalize its heat to assure that each vertex has the same ability of diffusing heat. Otherwise, vertices with high connectivity will diffuse heat more easily, which would suppress experts since experts usually have high connectivity. "Connectivity" does not necessarily mean "degree." For example, we can define the connectivity of a person as the number of distinct people who co-occur with him/her. Details of the choice of $d'(v)$ will be discussed in Section 4.1.

Similarly, for each word $i \in V_w$, the amount of heat it receives from neighboring vertices in a small time period Δt starting from t is

$$\begin{aligned} & f_i^w(t+\Delta t) - f_i^w(t) \\ & = \sum_{e\in E} H_w(i,e)\frac{w(e)}{\delta_w(e)} \sum_{j\in V_w} H_w(j,e)[\frac{f_j^w(t)}{d'(j)} - \frac{f_i^w(t)}{d'(i)}]\gamma_{ww}\Delta t \\ & + \sum_{e\in E} H_w(i,e)\frac{w(e)}{\delta_p(e)} \sum_{j\in V_p} H_p(j,e)[\frac{f_j^p(t)}{d'(j)} - \frac{f_i^w(t)}{d'(i)}]\gamma_{pw}\Delta t. \end{aligned} \tag{5}$$

Eq. (4) can be transformed as follows

$$\begin{aligned} & f_i^p(t+\Delta t) - f_i^p(t) \\ & = \gamma_{pp}\Delta t \Bigg(\sum_{e\in E}\sum_{j\in V_p} f_j^p(t) \frac{H_p(i,e)H_p(j,e)w(e)}{\delta_p(e)d'(j)} \\ & - \frac{f_i^p(t)}{d'(i)} \sum_{e\in E} H_p(i,e)w(e) \Bigg) \\ & + \gamma_{pw}\Delta t \Bigg(\sum_{e\in E}\sum_{j\in V_w} f_j^w(t) \frac{H_p(i,e)H_w(j,e)w(e)}{\delta_w(e)d'(j)} \\ & - \frac{f_i^p(t)}{d'(i)} \sum_{e\in E} H_p(i,e)w(e) \Bigg) \\ & = \gamma_{pp}\Delta t \Bigg(\sum_{j\in V_p} f_j^p(t) \sum_{e\in E} \frac{H_p(i,e)H_p(j,e)w(e)}{\delta_p(e)d'(j)} - f_i^p(t)\frac{d(i)}{d'(i)} \Bigg) \\ & + \gamma_{pw}\Delta t \Bigg(\sum_{j\in V_w} f_j^w(t) \sum_{e\in E} \frac{H_p(i,e)H_w(j,e)w(e)}{\delta_w(e)d'(j)} - f_i^p(t)\frac{d(i)}{d'(i)} \Bigg) \end{aligned}$$

Similarly, we can transform Eq. (5) as

$$\begin{aligned} & f_i^w(t+\Delta t) - f_i^w(t) \\ & = \gamma_{ww}\Delta t \Bigg(\sum_{j\in V_w} f_j^w(t) \sum_{e\in E} \frac{H_w(i,e)H_w(j,e)w(e)}{\delta_w(e)d'(j)} - f_i^w(t)\frac{d(i)}{d'(i)} \Bigg) \\ & + \gamma_{pw}\Delta t \Bigg(\sum_{j\in V_p} f_j^p(t) \sum_{e\in E} \frac{H_w(i,e)H_p(j,e)w(e)}{\delta_p(e)d'(j)} - f_i^w(t)\frac{d(i)}{d'(i)} \Bigg) \end{aligned}$$

We define an augmented vector $\mathbf{f}(t) = [(\mathbf{f}^p(t))^T\,(\mathbf{f}^w(t))^T]^T$. Let $\mathbf{W}_e$ denote the diagonal matrix containing edge weights in its main diagonal. Let $\mathbf{D}_p$ and $\mathbf{D}_w$ be diagonal matrices containing vertex degrees corresponding to people and words, respectively. Let $\mathbf{D}_{ep}$ and $\mathbf{D}_{ew}$ represent diagonal matrices containing degrees of hyperedges with respect to V_p and V_w, respectively. Let $\mathbf{D}_{p'}$ and $\mathbf{D}_{w'}$ represent diagonal matrices containing normalization terms for people and words, respectively. Notice that $\sum_{e \in E} H_p(i,e)H_p(j,e)w(e)$, $\sum_{e \in E} H_w(i,e)H_w(j,e)w(e)$ and $\sum_{e \in E} H_p(i,e)H_w(j,e)w(e)$ are the (i,j)-th elements of matrices $\mathbf{H}_p\mathbf{W}_e\mathbf{H}_p^T$, $\mathbf{H}_w\mathbf{W}_e\mathbf{H}_w^T$ and $\mathbf{H}_p\mathbf{W}_e\mathbf{H}_w^T$, respectively. Combining all $i \in V_p$, we can represent Eq. (4) in matrix-vector form:

$$\mathbf{f}^p(t+\Delta t) - \mathbf{f}^p(t) = \left[\begin{array}{cc} \mathbf{L}_{pp} & \mathbf{L}_{pw} \end{array} \right] \mathbf{f}(t)\Delta t, \tag{6}$$

where

$$\mathbf{L}_{pp} = \gamma_{pp}\mathbf{H}_p\mathbf{W}_e\mathbf{D}_{ep}^{-1}\mathbf{H}_p^T\mathbf{D}_{p'}^{-1} - (\gamma_{pp}+\gamma_{pw})\mathbf{D}_p\mathbf{D}_{p'}^{-1}, \tag{7}$$

and

$$\mathbf{L}_{pw} = \gamma_{pw}\mathbf{H}_p\mathbf{W}_e\mathbf{D}_{ew}^{-1}\mathbf{H}_w^T\mathbf{D}_{w'}^{-1}. \tag{8}$$

Similarly, we can compute $\mathbf{f}^w(t+\Delta t) - \mathbf{f}^w(t)$ as follows

$$\mathbf{f}^w(t+\Delta t) - \mathbf{f}^w(t) = \left[\begin{array}{cc} \mathbf{L}_{wp} & \mathbf{L}_{ww} \end{array} \right] \mathbf{f}(t)\Delta t, \tag{9}$$

where

$$\mathbf{L}_{wp} = \gamma_{pw}\mathbf{H}_w\mathbf{W}_e\mathbf{D}_{ep}^{-1}\mathbf{H}_p^T\mathbf{D}_{p'}^{-1}, \tag{10}$$

and

$$\mathbf{L}_{ww} = \gamma_{ww}\mathbf{H}_w\mathbf{W}_e\mathbf{D}_{ew}^{-1}\mathbf{H}_w^T\mathbf{D}_{w'}^{-1} - (\gamma_{ww}+\gamma_{pw})\mathbf{D}_w\mathbf{D}_{w'}^{-1}. \tag{11}$$

Combining Eq. (6) and (9), and letting $\Delta t \rightarrow 0$, finally we obtain the differential equation for $\mathbf{f}(t)$:

$$\frac{d}{dt}\mathbf{f}(t) = \mathbf{L}\mathbf{f}(t), \tag{12}$$

where $\mathbf{L}$ has the following block structure:

$$\mathbf{L} = \left[\begin{array}{cc} \mathbf{L}_{pp} & \mathbf{L}_{pw} \\ \mathbf{L}_{wp} & \mathbf{L}_{ww} \end{array} \right]. \tag{13}$$

Solving Eq. (12), we obtain

$$\mathbf{f}(t) = e^{t\mathbf{L}}\mathbf{f}(0), \tag{14}$$

where $\mathbf{f}(0) = [(\mathbf{f}^p(0))^T \; (\mathbf{f}^w(0))^T]^T$. Especially, we have

$$\mathbf{f}(1) = e^{\mathbf{L}}\mathbf{f}(0). \tag{15}$$

The exponential of a square matrix $\mathbf{L}$ is defined as

$$e^{\mathbf{L}} = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{L}^k. \tag{16}$$

In practice, it is difficult to obtain the exact value of $e^{\mathbf{L}}$. Therefore, a discrete approximation is used and Eq. (15) becomes

$$\mathbf{f}(1) = (\mathbf{I} + \frac{\mathbf{L}}{n})^n\mathbf{f}(0). \tag{17}$$

To determine the parameter n, Yang *et al.* proposed a heuristic method which chooses n so that the difference between the eigenvalues of $(\mathbf{I} + \frac{\mathbf{L}}{n})^n$ and $e^{\mathbf{L}}$ is less than a threshold [37]. In this paper we also employ this heuristic method to find proper values of n. In experiments, we find 100 iterations is usually sufficient for achieving good performance.

3.3 Interpretation of the Model

By constructing the matrix $\mathbf{L}$, we intrinsically aggregate the co-occurrence information among people and words in different Web pages to reflect the connection strength between each pair of objects. *This aggregation could be helpful for dealing with noises on the Web.* After the construction of $\mathbf{L}$, we propagate heat from query keywords (i.e. Eq. (17)) on this aggregated structure. Intuitively, names having strong connection not only with query keywords but also with other related names and words will be ranked high.

4 Diffusion for Expert Search

In this section we study how to apply the proposed diffusion model to our expert search problem.

4.1 Normalization Design

In Eq. (4) and (5), the heat normalization term $d'(v)$ assures that each vertex has the same ability to diffuse heat. Furthermore, we can also use $d'(v)$ to emphasize those vertices which we deem important (i.e. heat will flow to the vertex more easily). Since our goal is to rank people (names), in the following we will focus on the design of $d'(v)$ for people. For words, we simply set $d'(v) = d(v)$.

Intuitively, an expert should expose himself/herself more frequently than non-experts. Therefore, we consider $d(v)$ as a factor in $d'(v)$ for a name. Another characteristic of experts is that they tend to co-occur with many different people on the Web, e.g., a professor would co-occur with many students and other professors. Thus, we should also count in the number of distinct co-occurring names for a name (denoted by $Co(i)$ for name i). The heat normalization term for name i is defined as $d'(i) = d(i)Co(i)$. Fig. 4 shows a simple toy problem which illustrates the effect of heat normalization for people. Suppose our query is w_1, and we want to rank four people a, b, c and d. Assume that the four pages have the same weight. Intuitively, we expect c and d to be ranked higher than a since they co-occur with more people than a. If we use $d(v)$ as the normalization term we get $\{a : 0.109, c : 0.109, d : 0.109, b : 0.055\}$ as the ranking result, while we can get $\{c : 0.135, d : 0.135, a : 0.106, b : 0.067\}$ when $d(v)Co(v)$ is used. To summarize, we define $d'(v)$ as

$$d'(v) = \left\{ \begin{array}{ll} d(v)(Co(v)+1) & v \in V_p \\ d(v) & v \in V_w \end{array} \right. . \tag{18}$$

Here $(Co(v)+1)$ is used to avoid zero normalization when a name never co-occurs with other names.

By preliminary experiments, we find that some popular names, e.g., Bill Gates, tend to be ranked high for a variety of queries. Since these names occur much more frequently than other names, their absolute degree of "connection" with a query topic is also

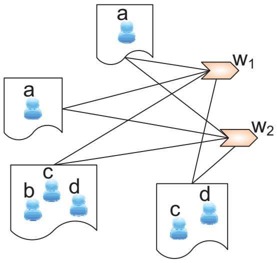


Fig. 4. A toy problem which illustrates the effect of heat normalization term $d'(v)$ for people.

likely to be high. However, if we consider their global occurrences, we can find they are actually connected with a variety of topics and the connection with each topic should be weakened. A similar analysis can be derived for words: general words tend to be related with a variety of topics and we should weaken their connection to each topic. To address this problem, we use a vertex's degree to normalize the weight of each edge from which it receives heat (called *global normalization*). Eq. (4) becomes:

$$
\begin{aligned}
& f_i^p(t+\Delta t) - f_i^p(t) \\
& = \sum_{e \in E} H_p(i,e) \frac{w(e)}{\sqrt{d(i)}\delta_p(e)} \sum_{j \in V_p} H_p(j,e)[\frac{f_j^p(t)}{d'(j)} - \frac{f_i^p(t)}{d'(i)}] \\
& * \gamma_{pp}\Delta t + \sum_{e \in E} H_p(i,e) \frac{w(e)}{\sqrt{d(i)}\delta_w(e)} \sum_{j \in V_w} H_w(j,e)[\frac{f_j^w(t)}{d'(j)} \\
& - \frac{f_i^p(t)}{d'(i)}]\gamma_{pw}\Delta t. \qquad (19)
\end{aligned}
$$

Eq. (5) is modified similarly. The reason that we adopt $\sqrt{d(i)}$ as the normalization term is that $d(i)$ can overly benefit names which only appear in a small number of related pages. Regarding the matrix-vector form, this corresponds to left-multiplying $\mathbf{D}_p^{-1/2}$ to $\mathbf{L}_{pp}$ and $\mathbf{L}_{pw}$, and left-multiplying $\mathbf{D}_w^{-1/2}$ to $\mathbf{L}_{ww}$ and $\mathbf{L}_{wp}$. Fig. 5 shows another toy problem which illustrates the idea of global normalization. w_1 is treated as the query. We can see although a co-occurs more frequently with w_1 than other people, he/she also appears in pages for other topics (w_2 and w_3). Therefore, we should suppress a in the ranking result. Without global normalization, the final ranking is $\{a : 0.328, c : 0.207, d : 0.207, b : 0.103, e : 0.069, f : 0.008\}$, while with global normalization the ranking is $\{c : 0.197, d : 0.197, a : 0.192, b : 0.124, e : 0.085, f : 0.004\}$.

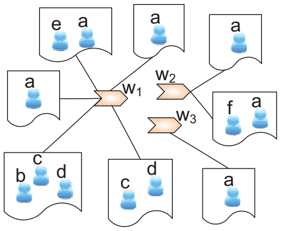


Fig. 5. A toy problem which illustrates the effect of global normalization.

```
Algorithm 1: Co-occurrence Diffusion
   Input: $\mathbf{H}_p$: weighted incidence matrix between people and pages;
          $\mathbf{H}_w$: weighted incidence matrix between words and pages;
          $\mathbf{W}_e$: diagonal matrix containing PageRank scores of pages; $\mathbf{f}$:
          the query vector; $\gamma_{pp}$, $\gamma_{ww}$, $\gamma_{pw}$: thermal conductivity between
          people, between words, between people and words, respectively
   Output: a ranked list of names according to the query
1  Model Construction
2     Compute the number of distinct co-occurring people $Co(i)$ for each
      person $i$ from $\mathbf{H}_p$
3     Construct degree matrices $\mathbf{D}_p$, $\mathbf{D}_w$, $\mathbf{D}_{ep}$, $\mathbf{D}_{ew}$ by $\mathbf{H}_p$, $\mathbf{H}_w$, $\mathbf{W}_e$
4     Construct heat normalization matrices $\mathbf{D}_{p'}$ by $\mathbf{D}_p$ and $Co(i)$'s, and
      $\mathbf{D}_{w'} = \mathbf{D}_w$
5     $\mathbf{L}_{pp} = \gamma_{pp}\mathbf{D}_p^{-\frac{1}{2}}\mathbf{H}_p\mathbf{W}_e\mathbf{D}_{ep}^{-1}\mathbf{H}_p^T\mathbf{D}_{p'}^{-1} - (\gamma_{pp}+\gamma_{pw})\mathbf{D}_p^{\frac{1}{2}}\mathbf{D}_{p'}^{-1}$
6     $\mathbf{L}_{pw} = \gamma_{pw}\mathbf{D}_p^{-\frac{1}{2}}\mathbf{H}_p\mathbf{W}_e\mathbf{D}_{ew}^{-1}\mathbf{H}_w^T\mathbf{D}_{w'}^{-1}$
7     $\mathbf{L}_{wp} = \gamma_{pw}\mathbf{D}_w^{-\frac{1}{2}}\mathbf{H}_w\mathbf{W}_e\mathbf{D}_{ep}^{-1}\mathbf{H}_p^T\mathbf{D}_{p'}^{-1}$
8     $\mathbf{L}_{ww} = \gamma_{ww}\mathbf{D}_w^{-\frac{1}{2}}\mathbf{H}_w\mathbf{W}_e\mathbf{D}_{ew}^{-1}\mathbf{H}_w^T\mathbf{D}_{w'}^{-1} - (\gamma_{ww}+\gamma_{pw})\mathbf{D}_w^{\frac{1}{2}}\mathbf{D}_{w'}^{-1}$
9     Construct $\mathbf{L}$ by $\mathbf{L}_{pp}$, $\mathbf{L}_{pw}$, $\mathbf{L}_{wp}$ and $\mathbf{L}_{ww}$
10 Diffusion and Ranking
11    for $k$ = 1 to $n$ do
12        $\mathbf{f} = (\mathbf{I} + \frac{\mathbf{L}}{n})\mathbf{f}$
13    end
14    Rank people names according to $\mathbf{f}$
```

4.2 Algorithm

The algorithm CoDiffusion is shown in Algorithm 1. It has two phases: "Model Construction" and "Diffusion and Ranking". In the Model Construction phase, we use the given data and parameters to construct matrix $\mathbf{L}$, which is then used in the Diffusion and Ranking phase to generate the ranked list of people names by iteratively multiplying the heat distribution vector $\mathbf{f}$ (line 12). Initially, $\mathbf{f}$ is set so that only elements corresponding to the query keywords equal to 1 and all other elements equal to 0.

4.3 Global Ranking vs. Local Ranking

There are two possible schemes to implement our algorithm: (1) we perform "Model Construction" on the entire Web collection and for each query we only need to perform the "Diffusion and Ranking" part in Algorithm 1. In other words, the first phase of Algorithm 1 needs to be done only once. Then the constructed model is used for all queries. We call this scheme *Global Ranking*; (2) we first obtain related Web pages for a query by querying the Web collection. Then we construct the model on the related pages and do diffusion. Regarding Algorithm 1, the input $\mathbf{H}_p$, $\mathbf{H}_w$ and $\mathbf{W}_e$ only contain entries for pages related to the query. Both phases are performed in an online fashion. We call this scheme *Local Ranking*. For Local Ranking, we cannot use the global normalization technique proposed in Section 4.1 since all the pages are related to the query. Therefore, we use $\sqrt{d(i)}$ of person i in the entire collection to normalize $\mathbf{f}^p$ directly.

Algorithm 2: One-Time Re-Ranking

```
Input: $\mathbf{H}_p$, $\mathbf{H}_w$, $\mathbf{W}_e$, $\gamma_{pp}$, $\gamma_{ww}$, $\gamma_{pw}$: as defined in Algorithm 1;
       $Top$: top $k$ names after the first run of CoDiffusion; $Scores$:
       corresponding ranking scores of the top $k$ names
Output: a ranked list of people names
Initialize query vector $\mathbf{f} = \mathbf{0}$
for $i$ = 1 to $k$ do
    $f_{Top(i)} = Scores(i)$
end
Invoke CoDiffusion without global normalization using parameters
$\mathbf{H}_p$, $\mathbf{H}_w$, $\mathbf{W}_e$, $\mathbf{f}$, $\gamma_{pp}$, $\gamma_{ww}$ and $\gamma_{pw}$
Return the ranked list generated by CoDiffusion
```

Compared to Local Ranking, Global Ranking could be more efficient for online ranking. However, in Global Ranking the algorithm can diffuse heat to partially relevant or even irrelevant pages, while Local Ranking can perform more focused diffusion. In Local Ranking, we can also compute more focused heat normalization term $d'(i)$ for people. Thus, Local Ranking could perform better than Global Ranking.

4.4 Algorithm Complexity

The major cost in CoDiffusion is incurred by matrix multiplications. Let n_p, n_w and n_e be the number of people names, words and pages, respectively. Suppose $\mathbf{H}_p$, $\mathbf{H}_w$ and $\mathbf{L}$ have m_p, m_w and m_l nonzero elements, respectively. For multiplication of two diagonal matrices, the time cost is linear in n_p, n_w or n_e. Multiplying $\mathbf{H}_p$ or $\mathbf{H}_w$ by a diagonal matrix cost $O(m_p)$ or $O(m_w)$, respectively. The dominant cost is due to $\mathbf{H}_p\mathbf{H}_p^T$, $\mathbf{H}_p\mathbf{H}_w^T$ and $\mathbf{H}_w\mathbf{H}_w^T$, where using the simple sparse matrix multiplication method [40] the corresponding time costs are $O(m_p n_p)$, $O(\min(m_p n_w, m_w n_p))$ and $O(m_w n_w)$, respectively. The cost of "Diffusion and Ranking" phase is $O(m_l n)$ where n is the number of iterations. The major space cost is the model matrix $\mathbf{L}$. When stored as a sparse matrix, the cost is $O(m_l + n_p + n_w)$. To give an intuition about how sparse the matrices are, the typical density of $\mathbf{H}_p$ is 0.003% and that of $\mathbf{L}$ is 0.38% in our experiments.

4.5 Refinement by Re-Ranking

The diffusion process employed in Algorithm 1 only sets query keywords as heat sources (i.e. queries). This can overly emphasize word-name diffusion and reduce the effect of name-name diffusion. Here we propose two re-ranking algorithms to refine the ranking results by setting top ranked people names as heat sources (i.e. queries), in order to boost reputable names for the query.

The first re-ranking algorithm is named *One-Time Re-Ranking*. The idea is that we set top k names from the ranking result generated by CoDiffusion as queries and invoke CoDiffusion (without global normalization) a second time. The intuition is that the top k names can be regarded as expert candidates and we could boost reputable experts by diffusing heat from these candidates. In the second re-ranking algorithm, we use an iterative process to gradually refine ranking results: initially we choose top k names from the result of CoDiffusion and use pages which contain at least two names in the top k names to build the diffusion model. Then we set these k names as queries and invoke CoDiffusion (without global normalization); in the j-th iteration we perform the same process with top $k - (j-1)k_0$ names from the last iteration, where k_0 is a small value (e.g. 50). By the second algorithm, we try to perform more and more focused diffusion in the community to find reputable experts. The second algorithm is named *Iterative Re-Ranking*. We summarize the two algorithms in Algorithm 2 and 3, respectively. For Iterative Re-Ranking, we discard names other than names in Top to better focus on top ranked names. We use the corresponding ranking scores outputted by CoDiffusion as query weights. In this way, the final ranking result will not deviate too much from the original one.

Algorithm 3: Iterative Re-Ranking

```
Input: $\mathbf{H}_p$, $\mathbf{H}_w$, $\mathbf{W}_e$, $\gamma_{pp}$, $\gamma_{ww}$, $\gamma_{pw}$: as defined in Algorithm 1;
       $Top$, $Scores$: as defined in Algorithm 2; $k_0$: deduction of $k$ in
       each iteration; $Iter\_num$: number of iterations
Output: a ranked list of people names
for $j$ = 1 to $Iter\_num$ do
    Initialize query vector $\mathbf{f} = \mathbf{0}$
    for $i$ = 1 to Length($Top$) do
        $f_{Top(i)} = Scores(i)$
    end
    Find pages containing at least two names in $Top$ and construct
    corresponding $\mathbf{H}'_p$, $\mathbf{H}'_w$ and $\mathbf{W}'_e$
    Invoke CoDiffusion without global normalization using
    parameters $\mathbf{H}'_p$, $\mathbf{H}'_w$, $\mathbf{W}'_e$, $\mathbf{f}$, $\gamma_{pp}$, $\gamma_{ww}$ and $\gamma_{pw}$
    Set $Top$ and $Scores$ to the top $k - j * k_0$ names and their
    corresponding scores outputted by CoDiffusion
end
Return a ranked list according to $Top$
```

4.5.1 Ambiguous Names

It is common that the same name can refer to different people. The global normalization technique proposed only considers the situation where all (or almost all) occurrences of a name refer to the same person, e.g. Bill Gates. It could hinder names which often refer to different people on the Web. For example, "Michael Jordan" can refer to a famous basketball player or a reputable professor in machine learning. By the re-ranking algorithms, we could find back those ambiguous names which are also reputable names for the query. In this paper, we concentrate on the problem of retrieving reputable names for a query based on ordinary Web pages. Certainly, as a preprocessing step, name disambiguation is helpful for our problem. However, it is a standalone ongoing research topic [1], [39], [35], [11] which is out of scope of this work.

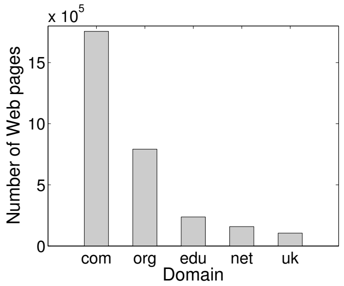


Fig. 6. Top five domains in our dataset.

5 Experiments

5.1 Data Preparation

Our experimental datasets were extracted from the ClueWeb09 Web collection which is a result of recent Web crawl and consists of about 1.04 billion Web pages in ten languages. We only considered the 500 million English Web pages. PageRank scores were computed based on the link graph among all the 500 million English Web pages. For people names, we extracted author names from the DBLP (Digital Bibliography & Library Project) bibliography dataset[3]. The reasons that we use DBLP author names are: (1) it contains a large number of names, ∼800K names; (2) it is easy to construct ground truth datasets for evaluation. The process for generating our experimental datasets is as follows: first we did a sequential scan through all the 500 million English Web pages to extract all the occurrences of author names, where simple rules, "First Middle Last" and "Last, First Middle", are used to find name occurrences. We discarded names which did not appear in those English pages. After this step we got 520,971 distinct people names and 37 million pages, each of which contains at least one person name. We extracted and processed those pages' text content and built index for them. Then we selected, from the remainder, the Web pages that contain at least 5 distinct people names and at least 30 distinct words, in order to reduce dataset size. This yields 3,608,265 pages and 478,896 names. Our task is to find top-10 or 20 among these names for a given query. To provide a notion of where these 3,608,265 pages come from, we show the top five domains of those pages in Fig. 6. We can see that a large amount of Web pages do not come from academic Websites (e.g. .edu). We formulated three datasets from these pages: (1) *DATA-3M*: which contained all 3,608,265 pages; (2) *DATA-1M*: which consisted of a random subset of 1 million pages from the 3,608,265 pages; and (3) *DATA-0.2M*: which consisted of a random subset of 200k pages from the 3,608,265 pages. We used DATA-1M for most of the experiments. DATA-3M and DATA-0.2M were used to investigate the influence of data sizes on the performance (Section 5.5).

5.2 Evaluation Methodology

We employ four baseline algorithms for performance comparison. The first two algorithms are simple heuristics which follow the intuition about topical experts discussed in Section 1. The first one, which is called *NameFreq*, computes the total number of times a name appears in pages that contain all the query keywords. Frequency in each page is weighted by the corresponding PageRank score. Thus, NameFreq actually computes the $d(i)$ for a person name i in a query-dependent local context. For NameFreq we also use $\sqrt{d(i)}$ to normalize the obtained ranking scores. The second one, *NameCoFreq*, counts the number of distinct names which co-occur with a name in pages containing all the query keywords. The third one is the language model based algorithm proposed in [2], which is one of the most prominent methods for organizational expert search, denoted by *LM*. The document-centric scheme is adopted. LM sorts people names by the probability of generating the query Q given the name i (i.e. $\Pr(Q|i)$), which marginalizes over all the documents associated with i. The last one, RW, is a random walk based approach proposed in [34] which performs random walks on a name-document bipartite graph. We adopt the finite random walk scheme since it showed good performance for organizational expert search. We also tried to use $\sqrt{d(i)}$ to normalize NameCoFreq, LM and RW, but the performance declines. The reason may be that ranking scores generated by those algorithms are not well correlated with $d(i)$.

TABLE 1
Three example queries from each of our two benchmark datasets.

Libra-GT	Manual-GT
Information Retrieval	Natural Language Processing
Machine Learning	Support Vector Machine
Algorithm and Theory	Reinforcement Learning

Two ground truth datasets are used to evaluate expert search algorithms. The first one is collected from Libra[4]. We crawl the Website to obtain the top 100 authors for each of the 24 research areas of computer science. The 24 area names are treated as test queries and the corresponding top 100 authors are taken as ground truth expert lists. This is reasonable since the top author lists are computed by structural bibliography data including the number of publications, citations and H-index, and we are trying to predict them from unstructured Web data. The other one is a manually labeled ground truth dataset used in [18], which contains 17 queries and the averaged number of experts for each query is 29.35. We refer to the two benchmark datasets as Libra-GT and Manual-GT, respectively. Libra-GT contains more general queries

[3] http://www.informatik.uni-trier.de/∼ley/db/

[4] http://libra.msra.cn/

while Manual-GT contains more specific ones. There are 41 queries in total. Table 1 shows some example queries. Three metrics are used for performance evaluation: Precision@n (P@n), Mean Average Precision and Normalized Discount Cumulative Gain (NDCG). P@n is the precision at rank n, which is defined as

$$\text{P@n} = \frac{\text{\# of relevant experts in top } n \text{ results}}{n}. \quad (20)$$

Average Precision (AP) is the average of precision scores after each correctly identified relevant expert:

$$\text{AP} = \frac{\sum_i \text{P@i} \times \text{corr}_i}{\text{\# of correctly identified relevant experts}}, \quad (21)$$

where $\text{corr}_i = 1$ if the person at position i is a relevant expert, otherwise $\text{corr}_i = 0$. MAP is the mean of average precision scores over all the test queries. NDCG at position n is defined as

$$\text{NDCG@n} = Z_n \sum_{i=1}^{n} (2^{r_i} - 1)/\log_2(i+1), \quad (22)$$

where r_i is the relevance rating of the person at rank i. In our case, r_i is 1 if the corresponding person is a relevant expert and 0 otherwise. Z_n is chosen so that the perfect ranking has a NDCG value of 1, i.e. all the relevant experts in the list are ranked at the highest positions. We investigate the top 20 results for each algorithm and report P@10, P@20, NDCG@10, NDCG@20 and MAP.

5.3 Performance Comparison

We compare CoDiffusion with the baseline algorithms. As aforementioned, we can implement CoDiffusion in two schemes: Global Ranking and Local Ranking. We can also run the baseline algorithms in these two schemes. In Global Ranking, we just run the algorithms on the entire dataset, while in Local Ranking, for each query we first search the index to get the set of related pages (which is a subset of the 1,080,259 pages in DATA-1M) and then run the algorithms on those related pages. We set $\gamma_{pp} = 700$, $\gamma_{pw} = 160$, $\gamma_{ww} = 2.5$ for CoDiffusion. How these parameters influence the performance will be explored in Section 5.4. The significance test in this subsection is based on all the 41 query topics from our two benchmark datasets.

TABLE 2
Performance comparison of expert search algorithms with respect to Libra-GT. Results for both Local Ranking and Global Ranking are reported. "N@n" is an abbreviation for NDCG@n.

Algorithm	P@10	P@20	MAP	N@10	N@20
Local Ranking:					
CoDiffusion	**.4125**	**.3625**	**.4977**	**.5220**	**.6816**
NameFreq	.1542	.1208	.2420	.3060	.4004
NameCoFreq	.1875	.1688	.3127	.3234	.4428
LM	.2125	.1833	.3344	.3635	.4863
RW	.1500	.1417	.2082	.2276	.3441
Global Ranking:					
CoDiffusion	**.1542**	**.1458**	**.2647**	**.2836**	**.4231**
NameFreq	.1083	.0979	.1843	.2268	.3386
NameCoFreq	.0875	.0813	.1459	.1824	.2661
LM	.0708	.0792	.1899	.2040	.3047
RW	.1250	.1292	.2458	.2515	.3961

TABLE 3
Performance comparison of expert search algorithms with respect to Manual-GT. Results for both Local Ranking and Global Ranking are reported. "N@n" is an abbreviation for NDCG@n.

Algorithm	P@10	P@20	MAP	N@10	N@20
Local Ranking:					
CoDiffusion	**.3765**	**.3088**	**.5031**	**.5387**	**.7066**
NameFreq	.1176	.1088	.1933	.2449	.3609
NameCoFreq	.1529	.1353	.2244	.2521	.3870
LM	.2176	.1824	.3773	.4353	.5644
RW	.1882	.1971	.3307	.3492	.5410
Global Ranking:					
CoDiffusion	.1176	**.1324**	.2178	.2031	.3564
NameFreq	.0647	.0794	.1363	.1432	.2767
NameCoFreq	.0882	.0853	.1812	.2052	.3285
LM	.1353	.1059	.2149	.2480	.3448
RW	**.1412**	.1265	**.4067**	**.4299**	**.5574**

The experimental results are shown in Tables 2 and 3, for Libra-GT and Manual-GT, respectively. We have the following observations. Firstly, our algorithm significantly outperforms the baseline algorithms in the context of Local Ranking (by t-test with $\alpha = 0.05$). NameFreq and NameCoFreq are simple heuristics and only use partial information. Although LM is a principled method, it treats people individually and therefore cannot capture the reputation knowledge contained in people co-occurrences. RW can capture reputation to some degree. However, it just relies on name-document bipartite relationships to propagate scores and does not explicitly model co-occurrence information. In Section 5.4, we will demonstrate that people co-occurrence information does contribute to the performance of CoDiffusion. By using a hpyer-graph model, our algorithm successfully leverages the co-occurrence information contained in Web pages to find the experts related to a query. Secondly, CoDiffusion does not show superior performance with the Global Ranking scheme, compared to LM and RW. The reason is that CoDiffusion treats each query keyword independently, which means heat can be diffused to partially relevant or even irrelevant pages. Our algorithm benefits a lot from Local Ranking since (1) we can perform more focused heat propagation in Local Ranking than in Global Ranking; (2) we can calculate more focused heat normalization scores for people. Therefore, Local Ranking is a better choice for CoDiffusion. Regarding efficiency, although we need to build the diffusion model for each query in Local Ranking, in practice CoDiffusion can still be faster in Local Ranking than in Global Ranking, since model scales are quite different (typically, each query only requires about 15000 relevant pages for model construction). We adopt the Local Ranking scheme for all the following experiments.

We show the top 10 names returned by CoDiffusion in Local Ranking for the query "Information Retrieval" in Table 4. It shows that most of these researchers are in the "top authors in Information Retrieval" provided by Libra. Note that the order of names does not totally conform to the ranking list in Libra since we make use of ordinary Web pages to rank authors. We would like to point out that among the 3,608,265 Web pages we use in experiments, there are only 22,567 coming from the DBLP Website. Two names are not in the Libra top author list. They are also IR researchers and co-occur frequently with senior IR researchers in our dataset. Consequently they also gain a lot of heat.

TABLE 4
Top 10 names returned by CoDiffusion for the query "Information Retrieval" in Local Ranking scheme.

Q: Information Retrieval	
Norbert Fuhr	Mounia Lalmas
Hsin-Hsi Chen	Jamie Callan
Alan F. Smeaton	Carol Peters
Gerard Salton	Saadia Malik
Iadh Ounis	W. Bruce Croft

5.4 Model Parameters

The proposed diffusion model has three parameters, i.e. γ_{pp}, γ_{ww} and γ_{pw}, which control the heat conductivity among people, among words and between people and words, respectively. To explore the influence of these parameters on the performance of CoDiffusion, we vary each parameter in turn and run our algorithm with Local Ranking scheme. When varying each parameter, the other two are fixed at 1. The results are averaged over all the 41 queries from Libra-GT and Manual-GT. Fig. 7 shows the plots. We report the performance in terms of P@10 and P@20. As one can see, the performance of our algorithm increases when increasing γ_{pp} (Fig. 7(a)) and γ_{pw} (Fig. 7(b)). We can interpret γ_{pp} and γ_{pw} as representing the importance of people co-occurrence information (reputation) and that of people-words co-occurrence information (relevance), respectively. This demonstrates that our observations about topical experts are effective in practice, i.e. an expert related to a query should co-occur frequently not only with query keywords, but also with many other people related to the query. Although we can improve further the performance by increasing γ_{pp} and γ_{pw}, we also need a larger n to get proper approximation of $e^{\mathbf{L}}$, which leads to more iterations in our algorithm (line 11-13 of Algorithm 1). This is a tradeoff between effectiveness and efficiency. We find the performance starts to decrease when γ_{pp} and γ_{pw} is set to a relatively large value (e.g. 5000), indicating there is a broad range of safe values. Regarding γ_{ww}, there is a performance increase at the early stage. The reason may be that semantically related words could help identifying experts. However, further increasing γ_{ww} can decrease the performance. This is intuitive since Web pages are noisy and some general words (e.g. polysemy) could gain more heat and blur the ranking results. In practice, we can perform cross-validation on benchmark datasets to select proper parameters.

5.5 Impact of Data Size

We investigate the impact of data sizes on search performance. Specifically, we run CoDiffusion (Local Ranking) on three datasets DATA-1M, DATA-3M and DATA-0.2M with the same parameter setting used in Section 5.3. Results are shown in Tables 5 and 6 for Libra-GT and Manual-GT, respectively. We find the situations are different for the two benchmark datasets. When increasing the dataset size (i.e. from DATA-1M to DATA-3M), the performance on queries in Manual-GT increases, while that on queries from Libra-GT decreases. This is because when data size grows, we see not only more co-occurrence evidences, but also more noises and ambiguous expertise evidences. Since most queries in Manual-GT are specific ones, the performance increase indicates we can obtain more useful co-occurrence information from DATA-3M than from DATA-1M. To the contrary, Libra-GT consists of very general research area names. Hence, we may already get enough co-occurrence information from DATA-1M. In DATA-1M, the averaged number of relevant pages for a query in Libra-GT is 16,188, while that for a query in Manual-GT is only 8,787. The results indicate that (1) for specific queries it is important that we obtain a large enough dataset in order to get enough co-occurrence information; (2) however, "the larger the better" is not the case for this general expert search problem. A solution could be that we first retrieve a moderate number of top relevant documents from a traditional search engine and run CoDiffusion.

TABLE 5
Performance comparison of CoDiffusion on different sizes of datasets with respect to Libra-GT. "N@n" is an abbreviation for NDCG@n.

Dataset	P@10	P@20	MAP	N@10	N@20
DATA-3M	.3935	.3563	.4864	.4742	.6769
DATA-1M	**.4125**	**.3625**	**.4977**	**.5220**	**.6816**
DATA-0.2M	.2935	.2643	.4872	.4858	.6658

On the other hand, when the dataset size is reduced (i.e. from DATA-1M to DATA-0.2M), the performance decreases dramatically for both benchmark datasets. This means our algorithm requires a large amount of co-occurrence information to achieve good performance. This is not the case for organizational expert search, where the data size is much smaller (e.g.

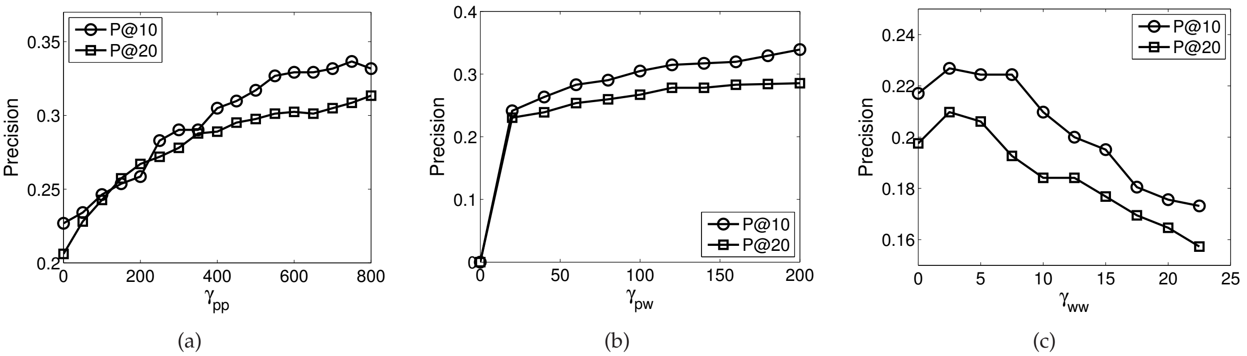


Fig. 7. Exploring the influence of three conductivity parameters (a) γ_{pp}, (b) γ_{pw} and (c) γ_{ww} on the performance of CoDiffusion. For each parameter, the other two parameters are fixed at 1. Results are averaged over all the 41 queries.

the W3C dataset has only 331,037 documents) and consequently there is not much people co-occurrence information. Therefore, our algorithm is not suitable for the traditional organizational expert search problem, where traditional methods (e.g. language model based methods) are better choices.

TABLE 6
Performance comparison of CoDiffusion on different sizes of datasets with respect to Manual-GT. "N@n" is an abbreviation for NDCG@n.

Dataset	P@10	P@20	MAP	N@10	N@20
DATA-3M	**.4000**	**.3352**	**.5088**	.5373	**.7083**
DATA-1M	.3765	.3088	.5031	**.5387**	.7066
DATA-0.2M	.2176	.1617	.4581	.5293	.6191

5.6 Beyond Research Queries

So far, we have used research related queries to evaluate our algorithm. However, unlike previous academic expert search based on bibliography data (e.g. [19]), we consider a more general expert search problem. Our algorithm can handle arbitrary queries as long as we have enough related pages. Hence, we show here some exploratory experimental results for queries which are irrelevant to academic research. Although our people names are from DBLP, there are many names which can refer to different people. Consequently, we have a lot of research irrelevant pages in the extracted pages from ClueWeb09. Table 7 shows the top 4 names returned by CoDiffusion for three queries: "USA Justice", "Basketball" and "Swimming". Our algorithm is able to retrieve senior people for the query topic at the highest positions. For "USA Justice", John Roberts is the current Chief Justice of the United States. John Paul should be referring to John Paul Stevens[5], who is a former Associate Justice of the Supreme Court of the United States. John Marshall is the 4th Chief Justice of the United States. Eric Holder[6] had joined the U.S. Justice Department and is the current Attorney General of the United States. For "Basketball", the names correspond to senior basketball players or coaches. Names returned for "Swimming" correspond to top swimmers who have won Olympic gold medals.

TABLE 7
Top 4 names returned by CoDiffusion for three queries which are irrelevant to academic research.

Q: USA Justice	Q: Basketball	Q: Swimming
John Roberts	Michael Jordan	Michael Phelps
John Paul	Abdul Jabbar	Ian Crocker
John Marshall	Tim Duncan	Gary Hall
Eric Holder	Dean Smith	Alain Bernard

It is not trivial to obtain the above ranking results. To demonstrate this, we show the top 4 names returned by the baseline algorithms for the query "Swimming" in Table 8. Clearly, these ranking lists are not as good as the ranking list generated by CoDiffusion, although they all put the most famous swimmer "Michael Phelps" at the top position. In particular, "Mike James" is a popular name and shows up relatively frequently in the related pages. Ana Ivanovic is a former World No. 1 tennis player[7]. However, we can also find many co-occurrences between her name and "Swimming". For example, her Wikipedia page says "she admitted that she trained in an abandoned swimming pool...". Although these two names appear frequently in related pages, they do not get into top 4 of NameCoFreq. "St. Thomas" appears frequently on the Web as an university name, although it is an author name in DBLP. "Juan Carlos" is a popular Spanish name and it also refers to different things (e.g., it is a part of a university[8] name). Hence, they co-occur with a lot of different names. Nevertheless, they do not get into the top 4 of NameFreq. Our

[5] http://en.wikipedia.org/wiki/John_Paul_Stevens

[6] http://en.wikipedia.org/wiki/Eric_Holder

[7] http://en.wikipedia.org/wiki/Ana_Ivanovi%C4%87

[8] http://www.urjc.es/

algorithm successfully makes use of different kinds of co-occurrence information to return top swimmers in the highest positions.

TABLE 8
Top 4 names by the baseline algorithms for "Swimming".

Q: Swimming			
NameFreq	**NameCoFreq**	**LM**	**RW**
Michael Phelps	Michael Phelps	Michael Phelps	Michael Phelps
Mike James	St. Thomas	St. Thomas	St. Thomas
Ian Crocker	Gary Hall	Gary Hall	Gary Hall
Ana Ivanovic	Juan Carlos	Ana Ivanovic	Ian Crocker

We also try to build quantitative evaluation of how well our algorithm does for arbitrary queries. We consider two queries "Nobel Physics" and "Apolo astronauts". For "Nobel Physics", we obtain names of Nobel laureates in Physics from http://nobelprize.org/nobel_prizes/physics/laureates/; for "Apolo astronauts" we get the list of all Apolo astronauts from Wikipedia. These names are treated as the ground truth. Our name set is extended to include all the ground truth names. Although adding ground truth names can lead to optimistic ranking results, it is fair for all the algorithms. The experimental results are shown in Table 9. As can be seen, for "Nobel Physics" almost all the algorithms can achieve good performance. NameFreq and LM do as good as CoDiffusion in terms of P@20. However, CoDiffusion has a better MAP, indicating it gives a better ranking. Regarding "Apolo astronauts", CoDiffusion performs much better than the baseline algorithms.

TABLE 9
Performance comparison of expert search algorithms on two queries: "Nobel Physics" and "Apolo astronauts".

Algorithm	**Nobel Physics**		**Apolo astronauts**	
	P@20	MAP	P@20	MAP
CoDiffusion	**.9500**	**.9537**	**.8500**	**.9511**
NameFreq	**.9500**	.9471	.3000	.4128
NameCoFreq	.5000	.7053	.2000	.6396
LM	**.9500**	.9396	.6500	.7669
RW	.8000	.9460	.7500	.8196

5.7 Effect of Re-Ranking

This subsection investigates the performance of two re-ranking algorithms proposed in Section 4.5. For both algorithms, we need to determine the number of top names we choose from the first run of CoDiffusion. Another important parameter for Iterative Re-Ranking is the number of iterations. We show their performance under different parameter values. In Iterative Re-Ranking, k_0 is set to 50.

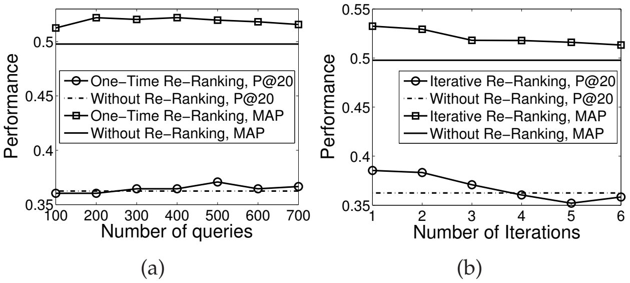


Fig. 8. The performance of (a) One-Time Re-Ranking and (b) Iterative Re-Ranking on Libra-GT.

Fig. 8(a) shows the performance of One-Time Re-Ranking on Libra-GT. We can see that One-Time Re-Ranking achieves the best performance when the number of name queries is around 500. Too few or too many name queries do not lead to good performance. Hence, for Iterative Re-Ranking we also set the initial name query number to 500 and vary the number of iterations. The experimental results are shown in Fig. 8(b). We find for most test instances in Libra-GT the best performance is achieved with one or two iterations. The reason may be that there are fewer and fewer Web pages when increasing the number of iterations. Lack of data is harmful to our method (Section 5.5). In Fig. 8 we also show the performance of the first run of CoDiffusion for comparison. We perform t-test with significance level $\alpha = 0.05$. Regarding P@20 and MAP, One-Time Re-Ranking does not show significant better performance, while Iterative Re-Ranking is significantly better than the first run of CoDiffusion. For P@10, the two re-ranking algorithms are as good as the first run of CoDiffusion. One-Time Re-Ranking and Iterative Re-Ranking have better NDCG@10, though the increases are not significant. Moreover, we would like to point out that for query "Machine Learning and Pattern Recognition" we can boost "Michael Jordan" from rank 208 to rank 2 by applying Iterative Re-Ranking. The two re-ranking algorithms do not show better performance on Manual-GT, which may be due to lack of data.

5.8 Running Time

We show in Fig. 9 the running time of CoDiffusion when varying the number of relevant pages in Local Ranking. The experiment is run on a PC with Intel Core i7 CPU and 12GB memory. The number of iterations is set to 100, which is sufficient for all the queries in our experiments. We can see the running time of CoDiffusion grows approximately linearly with the number of relevant pages. This is consistent with our complexity analysis in Section 4.4. Varying the number of Web pages only changes the number of nonzero elements in $\mathbf{H}_p$, $\mathbf{H}_w$ and $\mathbf{L}$. Diffusion and Ranking costs more time than Model Construction. This is because $\mathbf{L}$ is not only larger, but also much denser (see Section 4.4) than $\mathbf{H}_p$ and $\mathbf{H}_w$. CoDiffusion cannot outperform the baseline algorithms in terms of running time. The running time of RW is shown in

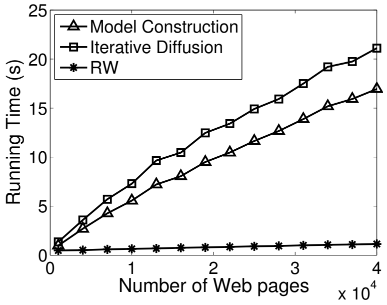


Fig. 9. Running time when varying the number of relevant Web pages (Local Ranking).

Fig. 9. RW is more efficient since its time cost depends on the number of nonzero elements in $\mathbf{H}_p$ which is much sparser than $\mathbf{L}$. We will discuss the scalability issue shortly. The running time of Global Ranking on DATA-1M are 190s and 245s, for Model Construction and Diffusion, respectively. This is because $\mathbf{L}$ with more pages is much denser. As shown in Section 5.3, with a typical number of 15K relevant pages, Local Ranking significantly outperforms Global Ranking. Thus, Local Ranking is a better choice in practice.

6 Discussions

Expert search on the Web is intrinsically different from enterprise expert search. As shown in Fig. 1 and Fig. 2, ordinary Web pages could be noisy and contain vague expertise evidences. The types of noises may not be limited to those given in Fig. 1. It is nearly impossible to do accurate entity and relation extraction in the Web setting. By using a large amount of co-occurrence information, noises could be suppressed since noisy co-occurrences would not appear frequently. The vague evidence issue can be alleviated by people co-occurrences: for the example in Fig. 2, Ana Ivanovic does not co-occur frequently with salient swimmers and therefore is not ranked high by our algorithm (Table 7).

This work's main focus is to *answer the following research question: whether we can retrieve experts for arbitrary topics from disparate contents and structures on the Web based on simple co-occurrence information*. We demonstrated that it was indeed feasible. The reason of using co-occurrence information is to avoid any complicated information extraction algorithm. The proposed diffusion model is general in that (1) association scores among people and words can be further adjusted by advanced techniques such as NLP through customizing the thermal conductivity for each pair of objects [10]; (2) other page quality measures [43] can be integrated through the hyperedge weighting scheme. However, we are not going to explore these possible enhancements in this work.

In the enterprise expert search, person identification is not difficult: we can obtain e-mail addresses or employee identifiers to uniquely identify an employee. The complete list of employees is known in advance. However, it is difficult to identify people on the Web as names are more available than email addresses. In this work, we generate a ranked list of people names and leave the person identification problem to users. With a returned name list, users can identify experts by searching their names together with the query topic through a Web search engine. We also use a set of names extracted from DBLP to bypass the name extraction problem, which is certainly an important research problem.

Scalability is important for Web scale problems. The Local Ranking method could be used for large scale Web expert search: we retrieve a moderate number (e.g. 20k) of top relevant pages from a traditional search engine and run CoDiffusion. The running time depends on the number of relevant pages, but not the size of the Web collection in the index. In our current implementation, we did not optimize the performance using multi-threading, multi-core, MapReduce or sampling techniques. There is room to further improve the running speed.

7 Conclusions

In this paper we studied a general expert search problem on the Web. We proposed not to deep-parse Web pages for expert search. Instead, it is possible to leverage co-occurrence relationships such as name-keyword co-occurrences and name-name co-occurrences to rank experts. A ranking algorithm called CoDiffusion was developed based on this concept. CoDiffusion adopts a heat diffusion model on heterogeneous hypergraphs to capture expertise information encoded in these co-occurrence relationships. Experiments on ClueWeb09 and two benchmark datasets consisting of research queries demonstrated that CoDiffusion outperformed the baseline algorithms significantly. Experiments on conductivity coefficients verified that co-occurrences were indeed useful. We also explored queries other than research related topics and showed that CoDiffusion could return good results and outperform baselines. Finally, we tried using re-ranking to boost performance.

8 Acknowledgments

This research was sponsored in part by the Army Research Laboratory under cooperative agreement W911NF-09-2-0053 (NS-CTA). D. Cai is supported by National Basic Research Program of China (973 Program) under Grant 2011CB302206. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

REFERENCES

[1] J. Artiles, J. Gonzalo, and S. Sekine, "Weps 2 evaluation campaign: overview of the web people search clustering task," in *2nd Web People Search Evaluation Workshop (WePS 2009)*, 2009.

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

[3] ——, "A language modeling framework for expert finding," *Information Processing & Management*, vol. 45, no. 1, pp. 1–19, 2009.

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

[5] K. Balog and M. de Rijke, "Finding similar experts," in *SIGIR*, 2007, pp. 821–822.

[6] K. Balog and M. De Rijke, "Associating people and documents," in *ECIR*, 2008, pp. 296–308.

[7] K. Balog and M. de Rijke, "Combining candidate and document models for expert search," in *TREC*, 2008.

[8] ——, "Non-local evidence for expert finding," in *CIKM*, 2008, pp. 489–498.

[9] K. Balog, P. Thomas, N. Craswell, I. Soboroff, P. Bailey, and A. P. de Vries, "Overview of the trec 2008 enterprise track," in *TREC*, 2008.

[10] H. Bao and E. Y. Chang, "Adheat: an influence-based diffusion model for propagating hints to match ads," in *WWW*, 2010, pp. 71–80.

[11] R. Bekkerman and A. McCallum, "Disambiguating web appearances of people in a social network," in *WWW*, 2005, pp. 463–470.

[12] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[13] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized non-negative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[14] D. Cai, X. Wang, and X. He, "Probabilistic dyadic data analysis with local and global consistency," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[15] P. R. Carlile, "Working knowledge: how organizations manage what they know," *Human Resource Planning*, vol. 21, no. 4, pp. 58–60, 1998.

[16] N. Craswell, A. P. de Vries, and I. Soboroff, "Overview of the trec 2005 enterprise track," in *TREC*, 2005.

[17] N. Craswell, D. Hawking, A. M. Vercoustre, and P. Wilkins, "P@noptic expert: Searching for experts not just for documents," in *Ausweb Poster Proceedings*, Queensland, Australia, 2001.

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

[19] ——, "Formal models for expert finding on dblp bibliography data," in *ICDM*, 2009, pp. 163–172.

[20] Y. Fang, L. Si, and A. P. Mathur, "Discriminative models of integrating document evidence and document-candidate associations for expert search," in *SIGIR*, 2010, pp. 683–690.

[21] Y. Fu, W. Yu, Y. Li, Y. Liu, M. Zhang, and S. Ma, "Thuir at trec 2005: Enterprise track," in *TREC*, 2005.

[22] D. Horowitz and S. D. Kamvar, "The anatomy of a large-scale social search engine," in *WWW*, 2010, pp. 431–440.

[23] M. Karimzadehgan and C. Zhai, "Constrained multi-aspect expertise matching for committee review assignment," in *CIKM*, 2009, pp. 1697–1700.

[24] M. Karimzadehgan, C. Zhai, and G. Belford, "Multi-aspect expertise matching for review assignment," in *CIKM*, 2008, pp. 1113–1122.

[25] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," in *ICML*, 2002, pp. 315–322.

[26] X. Liu, W. B. Croft, and M. Koll, "Finding experts in community-based question-answering services," in *CIKM*, 2005, pp. 315–316.

[27] X. Liu, Z. Nie, N. Yu, and J. R. Wen, "Biosnowball: automated population of wikis," in *SIGKDD*, 2010, pp. 969–978.

[28] H. Ma, H. Yang, M. R. Lyu, and I. King, "Mining social networks using heat diffusion processes for marketing candidates selection," in *CIKM*, 2008, pp. 233–242.

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

[30] ——, "Expertise drift and query expansion in expert search," in *CIKM*, 2007, pp. 341–350.

[31] D. Mimno and A. McCallum, "Expertise modeling for matching papers with reviewers," in *SIGKDD*, 2007, pp. 500–509.

[32] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," in *WWW*, 2006.

[33] P. Serdyukov and D. Hiemstra, "Being omnipresent to be almighty: The importance of the global web evidence for organizational expert finding," in *Proceedings of the SIGIR 2008 Workshop on Future Challenges in Expertise Retrieval (fCHER)*, 2008, pp. 17–24.

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

[35] J. Tang, A. Fong, B. Wang, and J. Zhang, "A unified probabilistic framework for name disambiguation in digital library," *IEEE Transactions on Knowledge and Data Engineering*, 2011.

[36] C. Yang, Y. Cao, Z. Nie, J. Zhou, and J. R. Wen, "Closing the loop in webpage understanding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 639–650, 2010.

[37] H. Yang, I. King, and M. R. Lyu, "Diffusionrank: a possible penicillin for web spamming," in *SIGIR*, 2007, pp. 431–438.

[38] D. Yimam-Seid and A. Kobsa, "Expert-finding systems for organizations: Problem and domain analysis and the demoir approach," *Journal of Organizational Computing and Electronic Commerce*, vol. 13, no. 1, pp. 1–24, 2003.

[39] M. Yoshida, M. Ikeda, S. Ono, I. Sato, and H. Nakagawa, "Person name disambiguation by bootstrapping," in *SIGIR*, 2010, pp. 10–17.

[40] R. Yuster and U. Zwick, "Fast sparse matrix multiplication," *ACM Transactions on Algorithms (TALG)*, vol. 1, no. 1, pp. 2–13, 2005.

[41] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: structure and algorithms," in *WWW*, 2007, pp. 221–230.

[42] D. Zhou, S. Orshanskiy, H. Zha, and C. Giles, "Co-ranking authors and documents in a heterogeneous network," in *ICDM*, 2007, pp. 739–744.

[43] J. Zhu, X. Huang, D. Song, and S. Rüger, "Integrating multiple document features in language models for expert finding," *Knowledge and Information Systems*, vol. 23, no. 1, pp. 29–54, 2010.

[44] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J. R. Wen, "Statsnowball: a statistical approach to extracting entity relationships," in *WWW*, 2009, pp. 101–110.

Ziyu Guan received the BS and PhD degrees in Computer Science from Zhejiang University, China, in 2004 and 2010, respectively. He is currently an assistant project scientist in the Information Network Academic Research Center (INARC) in the computer science department of University of California at Santa Barbara. His research interests include attributed graph mining and search, machine learning, expertise modeling and retrieval, and recommender systems.

Gengxin Miao is a PhD candidate at the ECE Department, UC Santa Barbara under the supervision of Prof. Louise Moser and Prof. Xifeng Yan. Prior to joining UCSB, she earned her BS and MS degrees from Tsinghua University. Her research interests spread in the fields of large-scaled data mining, statistical machine learning, social network analysis, information extraction, and information integration. Her PhD thesis investigates large-scale unstructured (and semi-structured) data within social networks to facilitate the information navigation and consumption with an emphasis on modeling the semantics.

Russell McLoughlin received his BS and MS degrees in Computer Science from the University of California at Santa Barbara. He is currently a computer scientist at Simigence Inc. Prior to Simigence, Russell held a position at the Biodefense Knowledge Center at Lawrence Livermore National Laboratory. His research interests include graph mining, machine learning and neuroanatomically based systems.

Xifeng Yan is an assistant professor at the University of California at Santa Barbara. He holds the Venkatesh Narayanamurti Chair in Computer Science. He received his Ph.D. degree in Computer Science from the University of Illinois at Urbana-Champaign in 2006. He was a research staff member at the IBM T. J. Watson Research Center between 2006 and 2008. He has been working on modeling, managing, and mining large-scale graphs in bioinformatics, social networks, information networks, and computer systems. He received NSF CAREER Award. He is a member of the IEEE.

Deng Cai is an Associate Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the PhD degree in computer science from University of Illinois at Urbana Champaign in 2009. Before that, he received his Bachelor's degree and a Master's degree from Tsinghua University in 2000 and 2003 respectively, both in automation. His research interests include machine learning, data mining and information retrieval. He is a member of the IEEE.