Co-Ocurrence-Based Diffusion for Expert Search on the Web

Ziyu Guan, Gengxin Miao, Russell McLoughlin, Xifeng Yan, Member, IEEE, and 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 webpages and thousands of names are considered. It has mainly two challenging issues: 1) webpages could be of varying quality and full of noises; 2) The expertise evidences scattered in webpages are usually vague and ambiguous. We propose to leverage the large amount of co-occurrence information to assess relevance and reputation of a person name for a query topic. The co-occurrence structure is modeled using 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-occurs with query keywords and co-occurs with 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 toward addressing the more general entity search problem without sophisticated NLP techniques.

Index Terms—Expert search, web mining, hypergraph, diffusion

1 INTRODUCTION

EXPERT search gained increasing attention from both industry and academia. The TREC enterprise tracks [14] boomed research work on organizational expert search [2], [8], [31], [32], [41], [18]. Variant expert search problems were also identified and addressed in other domains such as question answering [24], online forums [39] and academic society [29], [17], [40].

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 [20]. 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). Therefore, 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 webpages 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 webpages could be of varying quality (e.g., spam [30]) 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 webpages 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 webpages saying she used to train in a swimming pool, though she is not an expert in swimming.

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 webpages 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 webpages.

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 in a heterogeneous hypergraph where webpages 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. Intrinsically, CoDiffusion aggregates evidences collected from different webpages. This aggregation could be a good remedy for noises in web data. CoDiffusion complements traditional language model-based methods [2], if it applies their relevance scores.

**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 employs entity extraction [34], [25] and relation extraction [42] techniques. The major technique used in search engines like Renlifang is to extract structural information about entities and their relationships by deep-parsing webpages. 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 toward addressing the more general entity 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 reranking based on name pseudo relevance feedback. Empirical results on the ClueWeb09 web collection 2 show that our method is able to outperform 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], [37], [33], [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 [13]. However, creating a knowledge base manually is time consuming and laborious. Therefore, automatic approaches have been developed for building people profiles [15], [36]. Expert search became a hot research area since the start of the TREC enterprise track [14] 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 data set 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 [41], person-person similarity [4], internal document structures that indicate people’s association with document content [6], query expansion and relevance feedback using people names [28], [7], nonlocal evidence [8], [31], proximity between occurrences of query words and people names [19], [3]. Besides language models, other methods have been proposed for organizational expert retrieval. Macdonald and Ounis proposed a method based on voting and data fusion techniques [27]. Serdyukov et al. modeled associations between people and documents as a bipartite graph.
and performed probabilistic random walks to find relevant experts [32]. Fang et al. proposed a relevance-based discriminative learning framework for expert search [18]. Many other methods for organizational expert search were proposed during TREC Enterprise tracks.

Two benchmark data sets, W3C [14] and CSIRO [9], are the focus of the above organizational expert search works, which are crawls of the websites of World Wide Web Consortium 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 webpages and people names. The major concern in organizational expert search is relevance, while in our case we also need to consider the reputation of a person. 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 webpages 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 [39]. Liu et al. studied expert finding in community-based question answering websites and treated it as an IR problem [24]. Mimno and McCallum used topic modeling to address the problem of matching papers with reviewers [29]. Later Karimzadehgan et al. addressed this review assignment problem based on matching of multiple aspects of expertise [22], [21]. Deng et al. explored using language modeling and a topic-based model for expert finding in the DBLP bibliography data [17]. Zhou et al. proposed co-ranking authors and their publications using coupled random walks [40].

Finally, our work is also related to heat diffusion on graphs. In real world, heat diffuses in a medium from positions with a higher temperature to that with a lower temperature. The idea of heat diffusion was extended to the discrete graph setting, with applications such as dimension reduction [12], classification [23], antispamming [35], social network marketing [26] and online advertisement matching [10]. These studies considered diffusion in homogeneous graphs. In this paper, we develop a diffusion model based on heterogeneous hypergraphs for our expert search problem.

3 Heat Diffusion on Heterogeneous Hypergraphs

3.1 Notations and Problem Formulation

In a hypergraph, each edge (called hyperedge) 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 problem setting, there are three types of objects: people (names), words, and webpages, denoted by $P$, $W$, and $D$, respectively. By the co-occurrence relationships among $P$ and $W$ established by webpages, we can construct a heterogeneous hypergraph $G_{P,W} = (V, E)$ where $V$ contains vertices representing all the people and words and each $e \in E$ corresponds to a webpage. A toy example is shown in Fig. 3. $w(e)$ is the PageRank score of $e$’s corresponding webpage. The problem is, given $P$, $W$, $G_{P,W}$ and query keywords from $W$, to rank $P$ 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 $H_e$ be a $|V| \times |E|$ weighted incidence matrix where an entry $H_{p}(v, e) = w_{v,e}$ if $v \in e$ ($v \in V_p$) and 0 otherwise. $H_e$ is defined similarly for $V_w$. $w_{v,e}$ 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

$$d(v) = \begin{cases} \sum_{e \in E} w(e)H_{p}(v, e) & v \in V_p \\ \sum_{e \in E} w(e)H_{w}(v, e) & v \in V_w. \end{cases} \tag{1}$$

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 $f_v^p(t)$ and $f_v^w(t)$ to be the heat of vertex $i \in V_p$ and that of vertex $j \in V_w$ at time $t$, respectively. Let $F^p(t)$ and $F^w(t)$ 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 $F^p(0)$ and $F^w(0)$. Then, the problem is to derive the heat distribution at time $t$ ($F^p(t)$ and $F^w(t)$) given an initial distribution at time 0 ($F^p(0)$ and $F^w(0)$). 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 [23], [35]:

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

where $f(x, t)$ represents the heat at position $x$ and at time $t$, $\nabla^2$ denotes the Laplacian operator, and $\gamma$ 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: $\gamma_{pp}$, $\gamma_{ww}$, and $\gamma_{pw}$ 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 $\Delta t$. 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

\[ f_{ij}^e(t) = \frac{w(e)}{d^2(i)} \gamma_{ij} \Delta t, \]

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 $d^2(i)/d^2(e)$ to normalize $w(e)$ in that if a webpage 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., $d^2(v)$) 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 $d^2(v)$ 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 $\Delta t$ starting from $t$ is

\[ f_{ij}^w(t + \Delta t) - f_{ij}^w(t) = \sum_{e \in E} H_w(i,e) \sum_{j \in V_p} H_p(j,e) \frac{f_{ij}^p(t)}{d^2(j)} \gamma_{wj} \Delta t \]

\[ + \sum_{e \in E} H_p(i,e) \sum_{j \in V_w} H_w(j,e) \frac{f_{ij}^w(t)}{d^2(i)} \gamma_{pw} \Delta t. \]

Equation (4) can be transformed as follows:

\[ f_{ij}^p(t + \Delta t) - f_{ij}^p(t) = \gamma_{pp} \Delta t \left( \sum_{e \in E} \sum_{j \in V_p} H_p(i,e) H_p(j,e) \frac{f_{ij}^p(t)}{d^2(j)} \delta_p(e) d^2(i) \right) \]

\[ - \frac{f_{ij}^p(t)}{d^2(i)} \sum_{e \in E} H_p(i,e) w(e) \]

\[ + \gamma_{pw} \Delta t \left( \sum_{e \in E} \sum_{j \in V_w} H_p(i,e) H_w(j,e) \frac{f_{ij}^w(t)}{d^2(j)} \delta_w(e) d^2(i) \right) \]

\[ - \frac{f_{ij}^w(t)}{d^2(i)} \sum_{e \in E} H_p(i,e) w(e). \]

Similarly, we can transform (5) as

\[ f_{ij}^w(t + \Delta t) - f_{ij}^w(t) = \gamma_{ww} \Delta t \left( \sum_{e \in E} \sum_{j \in V_w} H_w(i,e) H_w(j,e) \frac{f_{ij}^w(t)}{d^2(j)} \delta_w(e) d^2(i) \right) \]

\[ - \frac{f_{ij}^w(t)}{d^2(i)} \sum_{e \in E} H_w(i,e) w(e) \]

\[ + \gamma_{pw} \Delta t \left( \sum_{e \in E} \sum_{j \in V_p} H_p(i,e) H_w(j,e) \frac{f_{ij}^w(t)}{d^2(j)} \delta_p(e) d^2(i) \right) \]

\[ - \frac{f_{ij}^w(t)}{d^2(i)} \sum_{e \in E} H_p(i,e) w(e). \]

We define an augmented vector $f(t) = [(f^p(t))^T (f^w(t))^T]^T$. Let $W_p$ denote the diagonal matrix containing edge weights in its main diagonal. Let $D_p$ and $D_w$ be diagonal matrices containing vertex degrees corresponding to people and words, respectively. Let $D_{pp}$ and $D_{ww}$ represent diagonal matrices containing degrees of hyperedges with respect to $V_p$ and $V_w$, respectively. Let $D_{pw}$ 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 $H_pW_pH_p^T$, $H_wW_wH_w^T$ and $H_pW_wH_p^T$, respectively. Combining all $i \in V_p$, we can represent (4) in matrix-vector form:

\[ f^p(t + \Delta t) - f^p(t) = [L_{pp} \quad L_{pw}]f(t) \Delta t, \]

where

\[ L_{pp} = \gamma_{pp} H_pW_pD_{pp}^{-1}H_p^T \quad (\gamma_{pp} + \gamma_{pw})D_pD_{pp}^{-1}, \]

and

\[ L_{pw} = \gamma_{pw} H_wW_pD_{pw}^{-1}H_p^T D_{ww}^{-1}. \]

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

\[ f^w(t + \Delta t) - f^w(t) = [L_{wp} \quad L_{ww}]f(t) \Delta t, \]

where

\[ L_{wp} = \gamma_{pw} H_wW_wD_{pw}^{-1}H_p^T D_{ww}^{-1}, \]

and

\[ L_{ww} = \gamma_{ww} H_wW_wD_{ww}^{-1}H_w^T D_{ww}^{-1}. \]
and

\[ L_{uv} = \gamma_{uu} H_u W_v D^{-1}_{uu} H^2_u D^{-1}_{uu} - (\gamma_{uv} + \gamma_{wu}) D_u D_w^{-1}. \]  

Combining (6) and (9), and letting \( \Delta t \to 0 \), finally we obtain the differential equation for \( f(t) \):

\[ \frac{d}{dt} f(t) = L f(t), \]  

where \( L \) has the following block structure:

\[ L = \begin{bmatrix} L_{pp} & L_{pu} \\ L_{up} & L_{uw} \end{bmatrix}. \]

Solving (12), we obtain

\[ f(t) = e^{Lt} f(0), \]  

where \( f(0) = [(f_p(0))^T (f_w(0))^T]^T \). Especially, we have

\[ f(1) = e^L f(0). \]  

The exponential of a square matrix \( L \) is defined as

\[ e^L = \sum_{k=0}^{\infty} \frac{1}{k!} L^k. \]

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

\[ f(1) = \left( I + \frac{L}{n} \right)^n f(0). \]

To determine the parameter \( n \), Yang et al. proposed a heuristic method which chooses \( n \) so that the difference between the eigenvalues of \( (I + \frac{L}{n})^n \) and \( e^L \) is less than a threshold [35]. 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

The intuition behind the diffusion model is as follows: by constructing the matrix \( L \), we intrinsically aggregate the co-occurrence information among people and words 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 \( L \), we propagate heat from query keywords (i.e., (17)) on this aggregated structure. Here, \( (Co(i) \) 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.

![A toy problem which illustrates the effect of heat normalization for people.](image)

Intuitively, an expert should expose himself/herself more frequently than nonexperts. 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; a senior forum user would actively answer questions for other users and consequently co-occurs with many different users. 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 \( w_k \), 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

\[ d'(v) = \begin{cases} d(v) (Co(v) + 1) & v \in V_p \\ d(v) & v \in V_w. \end{cases} \]

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 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 to 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). Equation (4) becomes

\[ f_p(t + \Delta t) - f_p(t) = \sum_{e \in E} H_p(i, e) \frac{w(e)}{\sqrt{d(i) d(e)}} \sum_{j \in V_p} H_p(j, e) \left[ \frac{f_p(t)}{d'(j)} - \frac{f_p(t)}{d'(i)} \right] \]

\[ \ast \gamma_{pp} \Delta t + \sum_{e \in E} H_p(i, e) \frac{w(e)}{\sqrt{d(i) d(e)}} \sum_{j \in V_p} H_w(j, e) \left[ f_w(t) \frac{f_p(t)}{d'(e)} - \frac{f_p(t)}{d'(e)} \right] \]

\[ \left[ \frac{f_p(t)}{d'(i)} \right] \gamma_{pw} \Delta t. \]

Equation (4) becomes

\[ \begin{align*}
\frac{df_p}{dt} + \frac{d}{dt} \gamma_{pp} \Delta t + \sum_{e \in E} H_p(i, e) \frac{w(e)}{\sqrt{d(i) d(e)}} \sum_{j \in V_p} H_p(j, e) \left[ \frac{f_p(t)}{d'(j)} - \frac{f_p(t)}{d'(i)} \right] \\
\ast \gamma_{pp} \Delta t + \sum_{e \in E} H_p(i, e) \frac{w(e)}{\sqrt{d(i) d(e)}} \sum_{j \in V_p} H_w(j, e) \left[ f_w(t) \frac{f_p(t)}{d'(e)} - \frac{f_p(t)}{d'(e)} \right] \\
- \left[ \frac{f_p(t)}{d'(i)} \right] \gamma_{pw} \Delta t.
\end{align*} \]
with global normalization the ranking is corresponding to left-multiplying of related pages. Regarding the matrix-vector form, this overly benefit names which only appear in a small number of related pages. Therefore, we should suppress names which only appear in a small number of related pages. Fig. 5 shows another toy problem which illustrates the idea of global normalization. 

Equation (5) is modified similarly. The reason that we adopt \( \sqrt{a(i)} \) 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 overly benefits names which only appear in a small number of related pages. Therefore, we should suppress names which only appear in a small number of related pages. Fig. 5 shows another toy problem which illustrates the idea of global normalization. 

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

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 \( L \), 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 \( f \) (line 12). Initially, \( f \) is set so that only elements corresponding to the query keywords equal to 1 and all other elements equal to 0.

Algorithm 1: Co-occurrence Diffusion

\[ f = \left( I + \frac{1}{m} D \right) f \]

4.3 Global Ranking versus 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 webpages 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 \( H_p, H_w \), and \( W \) 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 4.4 since all the pages are related to the query. Therefore, we use \( \sqrt{a(i)} \) of person \( i \) in the entire collection to normalize \( f \) directly.

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 \( m_p, m_w \), and \( m_e \) be the number of people names, words, and pages, respectively. Suppose \( H_p, H_w, \) and \( L \) have \( m_p, m_w \), and \( m_e \) nonzero elements, respectively. For multiplication of two diagonal matrices, the time cost is linear in \( m_p, m_w \), and \( m_e \). Multiplying \( H_p \) or \( H_w \) by a diagonal matrix costs \( O(m_p) \) or \( O(m_w) \), respectively. The dominant cost is due to \( H_p H_p^T \), \( H_p H_w^T \), and \( H_w H_w^T \), where \( H_p \) and \( H_w \) are the sparse simple matrix multiplication method [38].

The cost of “Diffusion and Ranking” phase is \( O(m) \) where \( m \) is the number of iterations. The major space cost is the model matrix \( L \). When stored as a sparse matrix, the cost is \( O(m_p + m_w) \). To give an intuition about how sparse the matrices are, the typical density of \( H_p \) is 0.003 percent and that of \( L \) is 0.38 percent in our experiments.

4.5 Refinement by Reranking

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 reranking 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 reranking 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 reranking 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 normal-
ization); 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
Reranking, we discard names other than names in $Top$ to
better focus on top ranked names. We use the correspond-
ning 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 2: One-Time Re-Ranking

1. Initialize query vector $f = 0$
2. for $i = 1$ to $k$
   1. Invoke CoDiffusion without global normalization using
   parameters $H_p$, $H_w$, $W_x$, $\gamma_p$, $\gamma_w$, and $\gamma_w$
2. Return a ranked list generated by CoDiffusion

Algorithm 3: Iterative Re-Ranking

1. Initialize query vector $f = 0$
2. for $i = 1$ to $\text{Length}(Top)$
   1. Invoke CoDiffusion without global normalization using
   parameters $H_p$, $H_w$, $W_x$, $\gamma_p$, $\gamma_w$, and $\gamma_w$
2. Return a ranked list generated 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 reranking 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
webpages. Certainly, as a preprocessing step, name
disambiguation is helpful for our problem. However, it is
a stand-alone ongoing research area [1], [37], [33], [11]
which is out of scope of this work.

5 EXPERIMENTS

5.1 Data Preparation

Our experimental data sets were extracted from the
ClueWeb09 web collection which is a result of recent web
crawl and consists of about 1.04 billion webpages in
10 languages. We only considered the 500 million English
webpages. PageRank scores were computed based on the
link graph among all the 500 million English webpages. For
people names, we extracted author names from the Digital
Bibliography & Library Project (DBLP) bibliography data
set. The reasons that we use DBLP author names are: 1) it
contains a large number of names, $\sim 800$ K names; 2) there
are both senior and junior people (i.e., experts and nonexperts);
3) it is easy to construct ground-truth data
sets for evaluation. The process for generating our experi-
mental data sets is as follows: first we did a sequential scan
through all the 500 million English webpages 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 webpages
that contain at least five distinct people names and at least
30 distinct words, in order to reduce data set 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
formulated three data sets 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 [32] 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 $d(i)$.

Two ground truth data sets are used to evaluate expert search algorithms. The first one is collected from Libra. 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 data set used in [16], which contains 17 queries and the averaged number of experts for each query is 29.35. We refer to the two benchmark data sets as Libra-GT and Manual-GT, respectively. Libra-GT contains more general queries 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 (AP), and Normalized Discount Cumulative Gain (NDCG). P@n is the precision at rank $n$, which is defined as

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

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

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

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

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

where $r_i$ is the relevance rating of the person at rank $i$. In our case, $r_i = 1$ 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 in Section 4.3, we can implement our algorithm in two schemes: Global Ranking and Local Ranking. We can also run the baseline algorithms in these two schemes. Here, we report the performance of both Global Ranking and Local Ranking. In Global Ranking, we just run the algorithms on the entire data set, 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 $\gamma_p = 700$, $\gamma_w = 160$, $\gamma_w = 2.5$ for CoDiffusion. How these parameters influence the performance will be explored in Section 5.4. The significance test in this section is based on all the 41 query topics from our two benchmark data sets.

The experimental results are shown in Tables 2 and 3, for Libra-GT and Manual-GT, respectively. We have the following observations. First, our algorithm significantly outperforms the baseline algorithms in the context of Local

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P@10</th>
<th>P@20</th>
<th>MAP</th>
<th>N@10</th>
<th>N@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Ranking:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoDiffusion</td>
<td>.4125</td>
<td>.3625</td>
<td>.4977</td>
<td>.5220</td>
<td>.6816</td>
</tr>
<tr>
<td>NameFreq</td>
<td>.1542</td>
<td>.1228</td>
<td>.2420</td>
<td>.3060</td>
<td>.4004</td>
</tr>
<tr>
<td>NameCoFreq</td>
<td>.1875</td>
<td>.1688</td>
<td>.3127</td>
<td>.3234</td>
<td>.4428</td>
</tr>
<tr>
<td>LM</td>
<td>.2125</td>
<td>.1833</td>
<td>.3344</td>
<td>.3635</td>
<td>.4863</td>
</tr>
<tr>
<td>RW</td>
<td>.1500</td>
<td>.1417</td>
<td>.2082</td>
<td>.2276</td>
<td>.3441</td>
</tr>
<tr>
<td>Global Ranking:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoDiffusion</td>
<td>.1542</td>
<td>.1458</td>
<td>.2647</td>
<td>.2836</td>
<td>.4231</td>
</tr>
<tr>
<td>NameFreq</td>
<td>.1083</td>
<td>.0979</td>
<td>.1843</td>
<td>.2268</td>
<td>.3386</td>
</tr>
<tr>
<td>NameCoFreq</td>
<td>.0873</td>
<td>.0813</td>
<td>.1459</td>
<td>.1824</td>
<td>.2661</td>
</tr>
<tr>
<td>LM</td>
<td>.0708</td>
<td>.0792</td>
<td>.1289</td>
<td>.2040</td>
<td>.3047</td>
</tr>
<tr>
<td>RW</td>
<td>.1250</td>
<td>.1292</td>
<td>.2458</td>
<td>.2513</td>
<td>.3961</td>
</tr>
</tbody>
</table>

Results for both Local Ranking and Global Ranking are reported. “N@n” is an abbreviation for NDCG@n.
Ranking (by t-test with $p = 0.05$). 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 hypergraph model, our algorithm successfully leverages the co-occurrence information contained in webpages to find the experts related to a query. Second, CoDiffusion does not show superior performance with 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 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 15,000 relevant pages for model construction). We adopt 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 webpages to rank authors. We would like to point out that among the 3,608,265 webpages 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 data set. Consequently, they also gain a lot of heat.

### 5.4 Model Parameters

The proposed diffusion model has three parameters, i.e., $\gamma_{pp}$, $\gamma_{pw}$, and $\gamma_{ww}$, 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 $\gamma_{pp}$ (Fig. 7a) and $\gamma_{pw}$ (Fig. 7b). We can interpret $\gamma_{pp}$ and $\gamma_{pw}$ 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

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>P@10</th>
<th>P@20</th>
<th>MAP</th>
<th>N@10</th>
<th>N@20</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Local Ranking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoDiffusion</td>
<td>.3765</td>
<td>.3088</td>
<td>.5031</td>
<td>.5387</td>
<td>.7066</td>
</tr>
<tr>
<td>NameFreq</td>
<td>.1176</td>
<td>.1088</td>
<td>.1933</td>
<td>.2249</td>
<td>.3609</td>
</tr>
<tr>
<td>NameCoFreq</td>
<td>.1529</td>
<td>.1353</td>
<td>.2244</td>
<td>.2521</td>
<td>.3870</td>
</tr>
<tr>
<td>LM</td>
<td>.2176</td>
<td>.1824</td>
<td>.3773</td>
<td>.4353</td>
<td>.5644</td>
</tr>
<tr>
<td>RW</td>
<td>.1882</td>
<td>.1971</td>
<td>.3307</td>
<td>.3492</td>
<td>.5410</td>
</tr>
<tr>
<td><strong>Global Ranking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoDiffusion</td>
<td>.1176</td>
<td>.1324</td>
<td>.2178</td>
<td>.2031</td>
<td>.3564</td>
</tr>
<tr>
<td>NameFreq</td>
<td>.0647</td>
<td>.0794</td>
<td>.1363</td>
<td>.1432</td>
<td>.2767</td>
</tr>
<tr>
<td>NameCoFreq</td>
<td>.0882</td>
<td>.0853</td>
<td>.1812</td>
<td>.2092</td>
<td>.3283</td>
</tr>
<tr>
<td>LM</td>
<td>.1353</td>
<td>.1099</td>
<td>.2149</td>
<td>.2480</td>
<td>.3448</td>
</tr>
<tr>
<td>RW</td>
<td>.1412</td>
<td>.1265</td>
<td>.4067</td>
<td>.4299</td>
<td>.5574</td>
</tr>
</tbody>
</table>

Results for both Local Ranking and Global Ranking are reported. “N@n” is an abbreviation for NDCG@n.

<table>
<thead>
<tr>
<th>Top 10 Names Returned by CoDiffusion for the Query “Information Retrieval” in Local Ranking Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Norbert Fuhr</td>
</tr>
<tr>
<td>Hsin-Hsi Chen</td>
</tr>
<tr>
<td>Alan F. Smeaton</td>
</tr>
<tr>
<td>Gerard Salton</td>
</tr>
<tr>
<td>Iadh Ounis</td>
</tr>
</tbody>
</table>
query keywords, but also with many other people related to the query. Although we can improve further the performance by increasing $\gamma_{pp}$ and $\gamma_{pw}$, we also need a larger $n$ to get proper approximation of $e^A$, which leads to more iterations in our algorithm (lines 11-13 of Algorithm 1). This is a tradeoff between effectiveness and efficiency. We find the performance starts to decrease when $\gamma_{pp}$ and $\gamma_{pw}$ is set to a relatively large value (e.g., 5,000), indicating there is a broad range of safe values. Regarding $\gamma_{uw}$, there is a performance increase at the early stage. The reason may be that semantically related words could help identifying experts. However, further increasing $\gamma_{uw}$ can decrease the performance. This is intuitive since webpages 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 data sets 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 data sets 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 data sets. When increasing the data set 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 data set 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 then run CoDiffusion.

On the other hand, when the data set size is reduced (i.e., from DATA-1M to DATA-0.2M), the performance decreases dramatically for both benchmark data sets. This means our algorithm requires a large amount of co-occurrence information to achieve good performance. While this is not the case for organizational expert search, where the data size is much smaller (e.g., the W3C data set 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.

### 5.6 Beyond Research Queries

So far, we have used research related queries to evaluate our algorithm. This is because we can easily obtain ground truth for those queries. However, unlike previous academic expert search based on bibliography data (e.g., [17]), 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. Notice that 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. Specifically, we adopt Local Ranking scheme: for a query we first search for relevant pages in our index and then run CoDiffusion on these pages. Table 7 shows the top four 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, who is a former Associate Justice of the Supreme Court of the United States. John Marshall is the fourth Chief Justice of the United States. Eric Holder had joined the US 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.

It is not trivial to obtain the above ranking results. To demonstrate this, we show the top four names returned by the baseline algorithms for the query “Swimming” in

---

### TABLE 5

Performance Comparison of CoDiffusion on Different Sizes of Data Sets with Respect to Libra-GT

<table>
<thead>
<tr>
<th>Dataset</th>
<th>P@10</th>
<th>P@20</th>
<th>MAP</th>
<th>N@10</th>
<th>N@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA-3M</td>
<td>3955</td>
<td>3563</td>
<td>4694</td>
<td>4742</td>
<td>6622</td>
</tr>
<tr>
<td>DATA-1M</td>
<td>4125</td>
<td>3625</td>
<td>4977</td>
<td>5220</td>
<td>6816</td>
</tr>
<tr>
<td>DATA-0.2M</td>
<td>2935</td>
<td>2643</td>
<td>4572</td>
<td>4558</td>
<td>6658</td>
</tr>
</tbody>
</table>

*“N@n” is an abbreviation for NDCG@n.*

### TABLE 6

Performance Comparison of CoDiffusion on Different Sizes of Data Sets with Respect to Manual-GT

<table>
<thead>
<tr>
<th>Dataset</th>
<th>P@10</th>
<th>P@20</th>
<th>MAP</th>
<th>N@10</th>
<th>N@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA-3M</td>
<td>4000</td>
<td>3352</td>
<td>5088</td>
<td>5373</td>
<td>7083</td>
</tr>
<tr>
<td>DATA-1M</td>
<td>3765</td>
<td>3088</td>
<td>5031</td>
<td>5387</td>
<td>7066</td>
</tr>
<tr>
<td>DATA-0.2M</td>
<td>2176</td>
<td>1617</td>
<td>4581</td>
<td>5293</td>
<td>6191</td>
</tr>
</tbody>
</table>

*“N@n” is an abbreviation for NDCG@n.*

---


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. 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 four 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 name). Hence, they co-occur with a lot of different names. Nevertheless, they do not get into the top four of NameFreq. Our algorithm successfully makes use of different kinds of co-occurrence information to return top swimmers in the highest positions.

We also try to build quantitative evaluation of how well our algorithm does for arbitrary queries. In particular, we consider two queries “Nobel Physics” and “Apolo astronauts.” For “Nobel Physics,” we obtain names of people who have won Nobel prizes 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.

### 5.7 Effect of Reranking

This section investigates the performance of two reranking 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 Reranking is the number of iterations. We show their performance under different parameter values. In Iterative Re-Ranking, \( k_0 \) is set to 50.

Fig. 8a shows the performance of One-Time Reranking on Libra-GT. We can see that One-Time Reranking 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 Reranking we also set the initial name query number to 500 and vary the number of iterations. The experimental results are shown in Fig. 8b. 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 webpages 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 Reranking does not show significant better performance, while Iterative Reranking is significantly better than the first run of CoDiffusion. For P@10, the two reranking algorithms are as good as the first run of CoDiffusion. One-Time Reranking and Iterative Reranking 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 Reranking. The two reranking 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 12 GB 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 webpages only changes the number
of nonzero elements in $H_p$, $H_w$, and $L$. Diffusion and Ranking costs more time than Model Construction. This is because $L$ is not only larger, but also much denser (see Section 4.4) than $H_p$ and $H_w$. CoDiffusion cannot outperform the baseline algorithms in terms of running time. The running time of RW is shown in Fig. 9. RW is more efficient since its time cost depends on the number of nonzero elements in $H_p$ which is much sparser than $L$. We will discuss the scalability issue shortly. The running time of Global Ranking on DATA-1M are 190 and 245 s, for Model Construction and Diffusion, respectively. This is because $L$ 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 Figs. 1 and 2, ordinary webpages 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) associations 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 [41] can also 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 e-mail addresses. In this work, we generate a ranked list of people’s name 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 multithreading, multicore, 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 webpages 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 data sets 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.

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 US Government. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein.

REFERENCES


Russell McLoughlin received the BS and MS degrees in computer science from the University of California at Santa Barbara in 2009 and 2010, respectively. He is currently a computer scientist in the Biodefense Knowledge Center at Lawrence Livermore National Laboratory. His research interests include graph mining, machine learning, and neuroanatomically motivated learning systems.

Xifeng Yan received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2006. He is an assistant professor at the University of California at Santa Barbara. He holds the Venkatesh Narayanamurti chair in Computer Science. 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 the US National Science Foundation (NSF) CAREER Award. He is a member of the IEEE.

Deng Cai received the bachelor’s and master’s degrees from Tsinghua University in 2000 and 2003, respectively, both in automation, and the PhD degree in computer science from University of Illinois at Urbana Champaign in 2009. He is an associate professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. His research interests include machine learning, data mining and information retrieval. He is a member of the IEEE.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.