

# Enhancing Expert Finding Using Organizational Hierarchies

Maryam Karimzadehgan<sup>1,†</sup>, Ryen W. White<sup>2</sup>, and Matthew Richardson<sup>2</sup>

<sup>1</sup> University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA  
[{mkarimz2@illinois.edu}](mailto:{mkarimz2@illinois.edu})

<sup>2</sup> Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA  
[{ryenw, mattri}@microsoft.com](mailto:{ryenw, mattri}@microsoft.com)

**Abstract.** The task in expert finding is to identify members of an organization with relevant expertise on a given topic. In existing expert finding systems, profiles are constructed from sources such as email or documents, and used as the basis for expert identification. In this paper, we leverage the organizational hierarchy (depicting relationships between managers, subordinates, and peers) to find members for whom we have little or no information. We propose an algorithm to improve expert finding performance by considering not only the expertise of the member, but also the expertise of his or her neighbors. We show that providing this additional information to an expert finding system improves its retrieval performance.

**Keywords:** Expert finding, organization hierarchy, graph smoothing.

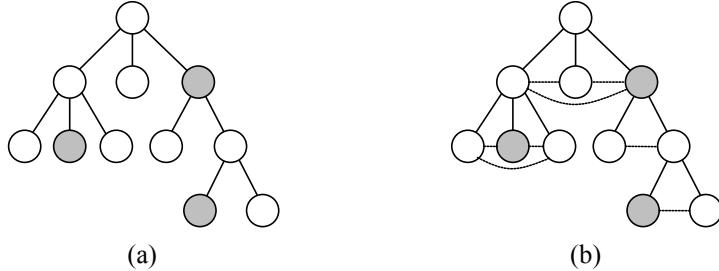
## 1 Introduction

The objective of an expert finding system is to help find people with the appropriate expertise to answer a question. This activity is particularly challenging in large organizations given the high number of employees and the degree of separation between inquirer and answerer, often in physical, divisional, and vocational senses.

For some questions, it can be difficult or impossible to find the answer using a Web search engine, especially for questions requiring tacit or procedural knowledge, or on topics internal to the organization. One common method for finding information in an organization is to use social connections, *i.e.*, ask people and follow referrals until finding someone with appropriate knowledge. However, this can be a time-consuming task, particularly in large, heterogeneous organizations such as Microsoft Corporation, where this research was conducted (with around 153,000 employees). Another technique is to send email to a discussion list (a mailing list used for discourse, often about a particular topic) or post to an online forum and await a response from an expert. Broadcasting a question can not only be unreliable, but also can unnecessarily interrupt too many people (if email notifications are involved), or have a high latency (if the inquirer must wait for an expert to read the forum posting). An attractive alternative is to direct the question to a small group of people, at least one of whom is expected to be an expert.

---

<sup>†</sup> Research performed during internship at Microsoft Research, Redmond, USA.



**Fig 1.** (a) A sample organizational hierarchy. Links between nodes denote management relationships. (b) Links are added between peers (members with the same manager). By propagating scores from members with profiles (gray nodes) to those without (white nodes), more members in the organization can be covered. Additionally, the scores of members with profiles can be refined.

Determining this set of people is known as the *expert finding* problem, and can be accomplished, for instance, by mining information about members of the organization, and then using this information as a basis for expert retrieval. One such source of information is a member’s email communications with discussion lists, particularly because many people use these lists to pose and answer questions. Other sources such as whitepapers or Web pages could be used, but often these have a lower coverage across members of the organization than email.

In this paper, we address the challenge of expert finding within organizations. In a similar way to the profile-centric method in Balog et al. [3], we use the content of members’ email to build an expertise profile for each of them. We also use the organizational hierarchy to improve retrieval through propagation. Figure 1 shows an example organizational hierarchy. The nodes represent employees and the links between them represent managerial reporting relationships. Two members are considered peers if they share the same direct manager. Reporting and peer relationships are represented by solid and dotted lines respectively.

The figure illustrates that hierarchy-based propagation allows an expert finding system to cover more employees and refine expertise scores. As we will demonstrate, those in close proximity to each other in the hierarchy tend to have similar topic knowledge, making the propagation of expertise scores among neighbors potentially beneficial. To the best of our knowledge, this is the first study that utilizes the organizational hierarchy to tackle the expert finding problem.

The evaluation of expert finding performance in large organizations is also challenging: it is unclear how to define the appropriate metrics to measure the quality of the retrieved experts given the scale involved and the impracticality of obtaining expert ratings for all members. Following standard Information Retrieval (IR) practice, we evaluate our algorithm using expert ratings from a sample of members, over a test set of queries, and report standard precision-recall metrics. We also experiment with an evaluation methodology not dependent on expert ratings, but instead based on predicting which member will answer a question posed to an email discussion list. Our findings demonstrate that using the organizational hierarchy to propagate expertise scores can improve the effectiveness of expert finding algorithms.

The remainder of the paper is structured as follows. We first review related work in Section 2 and motivate the use of the organizational hierarchy in Section 3. In Section 4, we propose our hierarchy-based algorithm. In Section 5, we describe our experi-

mental design and our evaluation measures. In Section 6, we present our experimental results, in Section 7 we discuss them, and we conclude in Section 8.

## 2 Related Work

Expert finding is a large and growing area with much previous work. Early work used standard Information Retrieval (IR) techniques to locate an expert on a given topic [1],[12],[27]. In these works, a person’s expertise was described as a term vector and the result was a list of related people. More recently, the Enterprise Track at the Text Retrieval Conference (TREC) was created to study expert finding. Participants in that track have investigated numerous methods, including probabilistic and language modeling techniques (*e.g.*, [3],[8],[21],[24]).

Since we use email discussion lists as a way to locate experts, our work is related to research on leveraging email documents for expert finding. Schwartz and Wood [23] were the first to identify groups of individuals with common interests. They used only email flows, not their contents. Their algorithm presented an unordered list of people related to a search query with little notion of relevance. ContactFinder [12] used the text and addresses of bulletin boards to identify experts. Xpertfinder [26] used a pre-existing hierarchy of subject areas, characterized by word frequencies, to identify experts in a specific area by analyzing the word frequencies of the email messages written by each individual. Xpertfinder did not rank the identified experts.

Page and Mueller [19] have shown that relying solely on word and document frequencies is limited. In order to overcome this problem, many systems [4],[5],[30],[31] use graph-based ranking algorithms, including HITS [14] and PageRank [20], in addition to content analysis, to locate experts. These graphs are built using the email correspondence between members, where each node is a member and directed edges point from email senders to recipients. Systems using graph-based algorithms effectively extract more information than is found in content only. One deficiency of these algorithms is that their performance depends significantly on characteristics of the network [31]. This makes such algorithms difficult to generalize to multiple expert finding contexts.

More recently-developed expert finding systems use social networks to help find experts; examples of such systems include MINDS [9] and ReferralWeb [7],[13]. These referral systems mimic human interaction by giving and following referrals. A referral system is a multi-agent system in which the agents cooperate by giving, pursuing, and evaluating referrals. MINDS emphasizes learning heuristics for referral generation whereas ReferralWeb targets bootstrapping the referral system.

Although our problem is related to the expert finding challenge addressed by these and similar systems, our setup is different. These approaches all assume that one can build a profile of all members of an organization, and find experts using those profiles. However, in large organizations it is unlikely that a reliable profile can be constructed for everyone: not everyone will send visible email or install an application capable of building profiles based on their email. It is important for the users of an expert finding system that it has access to a large pool of experts. More experts mean greater topic coverage and increased likelihood of a question being answered. The organizational hierarchy offers a way to handle sparseness by propagating expertise

scores from members with profiles to those without. It also allows us to refine expertise scores by propagating scores among those with profiles.

Our work is also related to graph smoothing. The need for smoothing originated from the zero count problem: when a term does not occur in a document, the maximum likelihood estimator would give it a zero probability. Smoothing is proposed to address this problem. While most smoothing methods utilize the global collection information with a simple interpolation approach [10],[18],[22],[29], other studies [6],[15],[16],[17],[25],[28] have shown that local corpus structures can be used to improve retrieval performance. A similar idea can be applied to expert finding: if we assume that people who are near each other in the organization will also have similar expertise, we can smooth a person's expertise score based on the scores of his or her neighbors. In the next section we test the validity of this assumption.

### 3 Motivation

In expert finding, we seek a set of individuals with expertise on a given topic ranked according to their estimated level of expertise. There are three basic tasks: (i) obtain an expert profile, (ii) find experts based on the profile, and (iii) evaluate the results. In our work, we assume that an expert can be represented by their email postings to discussion lists and focus on the second and third tasks.

At the outset of our studies we wished to determine if there was any value in utilizing the organizational hierarchy for expert finding. Previous work (*e.g.*, [2]) has suggested that those in close proximity within an organization are more likely to share knowledge via email. Our premise was that propagating expertise scores among neighbors (*e.g.*, managers, subordinates, and peers) in an organization would improve retrieval performance. To validate this premise we conducted a study within Microsoft Corporation. There are around 153,000 members of the organization, including temporary employees and vendors. A number of employees participate in a variety of topical discussions via internal email discussion lists. By crawling these lists, we were able to create expert profiles for 24% of all people in the organization<sup>1</sup>. We randomly selected the following question posed to one discussion list, where employees seek answers to work-related questions:

**Subject:** Standard clip art catalog or library  
**Body:** Do we have a corporate standard collection of  
clip art to use in presentations, specs, etc.?

The subject of the question was used as the query issued to the baseline expert finding system described in the next section. We contacted the retrieved employees and asked them to rate their expertise in answering this question on the following scale:

- 0 = I wouldn't know where to look to get the answer
- 1 = I could half-answer, point to someone who would know, or know a bit about it
- 2 = I can answer it

---

<sup>1</sup> This demonstrates the extent of the problem we are trying to address. If traditional expert finding algorithms were used, 76% of the company would be excluded from consideration as a potential expert. The use of hierarchy-based propagation helps address this challenge.

68 employees provided their expert rating for this question. We then identified the 632 employees situated at most one step in the organizational hierarchy from these 68 employees (*i.e.*, direct managers, direct subordinates, and peers) and asked them to also rate their ability to help answer the question posed to the original experts. 146 (23.1%) responded to this request and the results are summarized in Table 1. The table shows, for a given expertise rating (“source rating”), the mean rating provided by neighbors of the employees with this level of expertise.

**Table 1.** Mean neighbor rating in relation to source member rating.

Source rating	Mean neighbor rating	<i>N</i>
0	0.45	46
1	0.86	39
2	1.41	61
<i>Average over all ratings</i>	<i>0.96</i>	<i>146</i>

As can be seen in the table, the source and neighbor ratings are correlated, supporting our premise that those in close proximity in the organization have similar knowledge, in terms of their ability to help answer a particular question. From this, it seems that the knowledge of a neighbor may be useful to refine our estimate of an employee’s knowledge, particularly for employees who we have little or no information about (*e.g.*, we can boost his or her expertise score to be more confident of his or her ability to answer a question).

We believe there are at least two reasons why neighboring employees are likely to have similar interests and expertise: (i) they may work on the same Microsoft product, or (ii) their role may be similar. Note that the question was not specific to a particular product, but still may tend to be known by employees of the same type (*e.g.*, product planners, as opposed to software testers) or same sub-organization (*e.g.*, someone in the sales organization vs. the legal department).

These findings demonstrate the potential of propagating expertise scores among neighbors in the organizational hierarchy. In the next section we describe our algorithm that leverages the hierarchy for this purpose.

## 4 Expert Finding

We state the problem of identifying candidates who are experts on a given topic based on the following:

$$p(e|q) = \frac{p(q|e)p(e)}{p(q)} \quad (1)$$

Where  $q$  is the topic (query) and  $e$  is the expert.

We rank experts according to this probability. The top  $k$  candidates are experts on a given topic. For the purposes of ranking experts for a given query,  $p(q)$  is the same for all experts. We also assume *a priori* that all members have equal probability of

being an expert, so  $p(e)$  is the same for all experts as well. With these assumptions, the expert ranking becomes:

$$p(e|q) \propto p(q|e) \quad (2)$$

#### 4.1 Baseline Algorithm

In order to determine  $p(q|e)$ , we adapt the generative probabilistic language modeling techniques from IR. We build a representation of the individual using the email associated with the person, and measure the probability that this model would generate the query. We use language modeling with Dirichlet-prior smoothing [29] as follows:

$$p(q|e_j) = \prod_{w \in q} \frac{c(w, e_j) + \mu p(w|E)}{N_{e_j} + \mu} \quad (3)$$

Where  $e_j$  is the text representation of expertise for the  $j^{th}$  expert,  $c(w, e_j)$  is the number of times word  $w$  occurs in  $e_j$ , and  $N_{e_j}$  is the total number of words in  $e_j$ . The background language model,  $p(w|E)$ , is estimated using the entire set of expertise documents  $E$ , and  $\mu$  is the Dirichlet prior smoothing to be set empirically.

#### 4.2 Hierarchy-Based Algorithm

The baseline method works only if we have email for all members of an organization. Since it is unlikely that we will have this information for all members, we propose using the organizational hierarchy as an additional data source.

The hierarchy-based algorithm works as follows. First, the employees are scored based on Equation 3. Then, their scores are locally smoothed with their neighbors' scores using the following equation:

$$p_{smooth}(q|e_j) = \alpha p(q|e_j) + \frac{(1 - \alpha)}{N_j} \sum_{i=1}^{N_j} p(q|e_i) \quad (4)$$

Where  $\alpha$  is a weighting parameter and  $N_j$  is the number of neighbors for employee  $e_j$ .  $p(q|e_j)$  and  $p(q|e_i)$  are the initial scores for the employee  $e_j$  and his neighbors,  $e_i$ , respectively, and both are calculated based on Equation 3. In multi-step propagation, the scores are computed by considering all neighbors which are up to two-steps (for two-level propagation) or three-steps (for three-level propagation) away from  $e_j$ , the source node.

### 5 Evaluation

In this section, we describe the experiments conducted to evaluate the effectiveness of the hierarchy-based expert finding algorithm. Our research question is whether the inclusion of the organization hierarchy improves the retrieval effectiveness of a state-

of-the-art expert finding algorithm. We conducted our experiments within Microsoft Corporation. This gave us the large-scale environment necessary to test our approach. We begin this section by describing the expert-rating data used as ground truth, the data used for expert profiling, and our evaluation methodology.

### 5.1 Expert-Rating Data

We gathered expert-rating data to compare the hierarchy-based algorithm and the baseline algorithm, both described in Section 4. We used an internal email discussion list that contains questions and answers on a broad range of subjects. The purpose of the list is for employees to ask miscellaneous questions when they do not know where else to turn to; postings to the list are typically relatively brief questions or answers (as opposed to other lists that are used for general discussion on a topic). The list includes technical questions (*e.g., Where can I get technical support for MS SQL Server?*), recruitment questions (*e.g., Who is the Microsoft representative for college recruiting at UT Austin?*), and logistical questions (*e.g., How do I obtain a conference call leader and participant passcode?*), among others. We randomly selected 20 questions from the thousands of questions posed to this list to serve as our test set. We created an online survey, where for each of the 20 questions we provided the email subject and body, and asked employees to rate themselves based on the three-point answer scale described earlier: 0 = *I wouldn't know where to look to get the answer*, 1 = *I could half-answer, point to someone who would know, or know a bit about it*, and 2 = *I can answer it*. We distributed the survey to the 1,832 members of the discussion list. In total, 192 (10.5%) list members responded to the call and provided expert ratings for all 20 questions. We removed three respondents from the data set who provided the same answer rating for all questions (all zeros). These individuals may not have been diligent with the completion of the survey, and removing them did not significantly affect our results. This gave us 189 experts with ratings across the 20 queries in the test set.

### 5.2 Expert Profiling

We used email sent to internal discussion lists within Microsoft as a source of information to build expert profiles (leaving out the discussion list used to build the expert-rating evaluation set described in Section 5.1). These emails are visible to all employees through a shared resource. Employees post questions to these lists and other employees offer answers. We attempted to build a profile for each person in the organization by considering the emails they sent to the list in reply to posted questions (we considered only the portion of the email that they wrote, not the content of the question itself). This resulted in approximately 36,000 profiles (covering around 24% of the company). The average number of emails used to build a member's profile is 29; the median is 6. We extracted free-text contents from each email and used the Krovetz stemmer [11] to stem the words in the text. We also removed stop words such as “a” and “the”.

### 5.3 Methodology

We compared our hierarchy-based algorithm (Section 4.2) with the baseline algorithm (Section 4.1) that does not use the organizational hierarchy, but is a sub-part of the hierarchy-based algorithm. This allowed us to directly test the effect of adding hierarchy information. We set the Dirichlet prior,  $\mu$ , to 100 and the smoothing parameter,  $\alpha$ , to 0.9 (see Section 7 for details on these parameter settings).

We used the email subjects of the 20 selected questions as test queries. Since the goal was to find people who could directly answer the question, we regarded an expert rating of 2 as relevant and a rating of 0 or 1 as non-relevant. For each query, we generated a ranked list of employees using each of the expert finding algorithms. We computed precision-recall curves for each question and averaged across all questions. In the next section we present our findings.

## 6 Findings

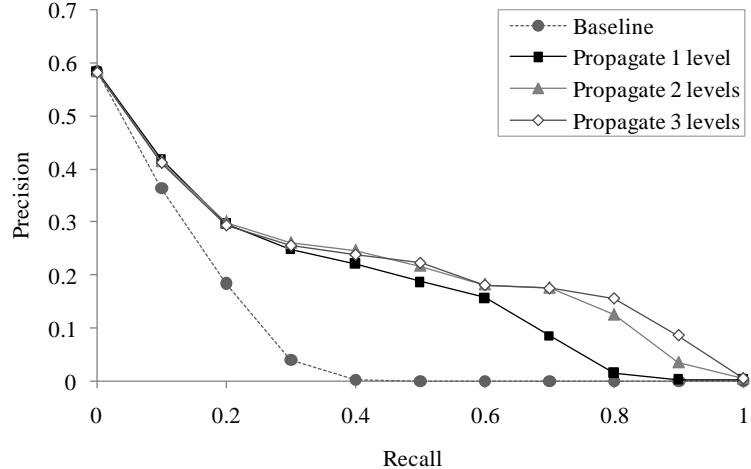
In this section we compare the retrieval effectiveness of our hierarchy-based algorithm with the baseline algorithm, both described in Section 4. We first evaluate the algorithms using the expert-rating data described in Section 5.1, and then describe an automatic evaluation method that does not require experts to rate themselves.

### 6.1 Expert-Rating Evaluation

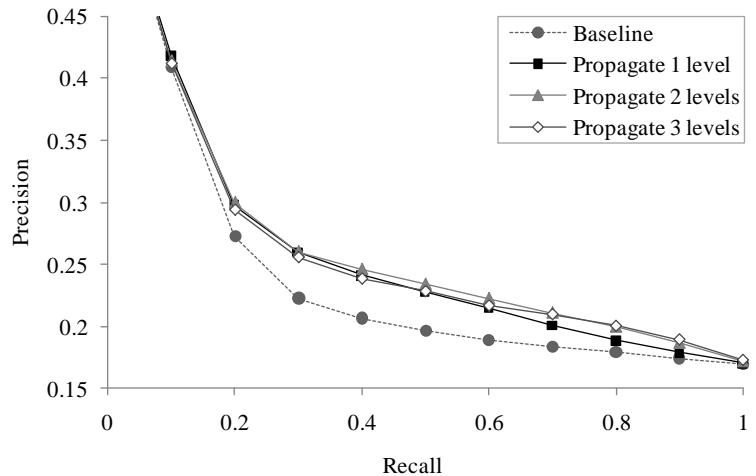
For each question, we used each algorithm to rank all employees in the organization. We then kept only those employees for which we had expert ratings (at most 189), maintaining their relative rank order. Figure 2 shows the interpolated average 11-point precision-recall curve for the baseline and the hierarchy-based algorithms. The figure also shows the precision-recall results for two- and three-level propagation. Note that the precision is zero at high recall since none of the algorithms retrieved all of the 189 prospective experts, and precision is defined to be zero for unattained recall levels. The algorithms have a similar precision at a recall level of zero. At each recall point above zero, the hierarchy-based algorithm (“Propagate 1 level”) outperforms the baseline. The results also show that two- and three-level propagation helps slightly at higher recall levels.

The effect observed in Figure 2 could be explained by the hierarchy-based algorithm simply returning more employees than the baseline (the one-, two-, and three-level propagation added on average 63, 87, and 99 employees per question). To verify that the hierarchy-based algorithm also *ranks* employees better, we conducted the following experiment.

As before, we used each algorithm to rank all employees, and kept only those for which we had expert ratings, maintaining their relative rank order. Unlike the previous approach we did not ignore rated employees that were not retrieved. Instead, we appended them to the end of the result list (giving them a retrieval score of zero) in random order so as to always have exactly 189 employees ranked by each algorithm for each query. We computed precision-recall curves for each expert finding algorithm, where each point was averaged across 100 runs (each with a random ordering



**Fig 2.** The average precision (averaged across 20 questions) vs. recall for both the baseline and hierarchy-based algorithms.



**Fig 3.** Average precision vs. recall for 20 questions and all 189 employees. The interpolated precision at zero for all algorithms was approximately 0.58. To aid exposition, we adjust the scale of the y-axis to highlight the differences at all other recall levels.

for non-retrieved employees). The curves are shown in Figure 3. The interpolated precision at zero for all algorithms was approximately 0.58. To aid exposition, we adjust the scale of the y-axis to highlight the differences at all other recall levels.

The results in Figure 3 show that the hierarchy-based algorithms are also better at ranking experts. We measured their statistical significance using a paired *t*-test at a significance level of 0.05. The difference between the hierarchy-based algorithm and the baseline is statistically significant at recall  $\geq 0.3$  (all  $t(19) > 2.37$ , all  $p < .03$ ). The

differences between both multi-level propagation algorithms and the baseline are statistically significant at recall  $\geq 0.2$  (all  $t(19) > 2.14$ , all  $p < .05$ ). Differences between single-level and multi-level propagation are significant at recall levels  $\geq 0.8$  (all  $t(19) > 2.83$ , all  $p \leq 0.01$ ). We also conducted experiments considering a rating of 1 or 2 as relevant and 0 as non-relevant; the results were qualitatively similar (the hierarchy-based algorithms continued to equal or outperform the baseline algorithm at all levels of recall).

## 6.2 Automatic Evaluation

Human judgments can be costly to obtain, especially for the large number of questions, and the variety of question types, required to thoroughly evaluate an expert finding algorithm. We experimented with alternative ways to evaluate our algorithms automatically without a manual judging effort. We devised a task whereby each algorithm was presented with a set of queries sent to an internal email discussion list and was asked to predict who in the company would answer the questions. Assuming that those who answer the questions are experts, a good expert finding algorithm should perform well at this task (note that the expert profiles were built ignoring this list).

We use a variant of mean reciprocal rank (MRR) as our evaluation metric. Since only a fraction of those who *could* answer the question actually *do*, MRR values will be very small. Thus, to aid exposition, we report an analog to inverse MRR, called *mean rank*, which is, for each question, the rank of the first actual answerer, averaged across all questions. A lower value indicates greater retrieval effectiveness.

For our test, we selected 600 random questions from the discussion list, and computed the mean rank for the baseline and hierarchy-based algorithms, which were 3039 and 798, respectively. From these scores, we can see that the hierarchy-based algorithm is better at ranking experts. Although these ranks seem high, the mean rank of an uninformed (random) algorithm would be approximately 60,000 (*i.e.*,  $average_q(153,000 / \# \text{answerers for question } q)$  ). The difficulty in the task lies in the fact that only a subset of those who *can* answer a question actually *did* (they may have been busy, or someone may have already answered). That said, this challenge affects both algorithms equally, so the measure can be used for algorithm comparison, and, as we have shown here, the results using mean rank correlate with the findings based on precision-recall.

## 7 Discussion and Future Work

The findings presented in the previous section demonstrate the value of organizational hierarchy-based propagation for expert finding. As part of our research we examined how some of the parameters used in baseline and hierarchy-based algorithms affect retrieval performance. We used both the expert-rating data and the mean rank measure, and obtained similar trends in the findings of each technique. We found that the baseline algorithm performance was insensitive to its Dirichlet prior smoothing parameter  $\mu$ , over a range from 10 to 10000. The best retrieval performance was achieved when  $\mu = 100$ . The hierarchy-based algorithm was also relatively insensitive to its

weighting parameter  $\alpha$ , for  $\alpha > 0.5$ . The best retrieval performance was achieved when  $\alpha = 0.9$ . Due to space limitations, we do not show the sensitivity plots.

Instead of propagating expertise *scores*, we also investigated propagating the *keywords* in the expert profiles to the neighbors and scoring employees based on these expanded profiles. The results were significantly worse than the score-based approach. Mei et al. [17] also found that keyword-based propagation does not perform as well as score-based propagation.

Our future work involves studying other expert finding algorithms and organizations to determine whether our findings hold for them. In addition, we plan to enhance the hierarchy-based algorithm, for instance by weighting the edges between individuals differently depending on their relationship (*e.g.*, a peer-to-peer relationship may differ from a manager-to-subordinate relationship), and study which relationships are most influential in improving retrieval performance. We will also experiment with propagating information such as whitepapers, personal websites, and communication patterns, to meet our goal of enabling rich modeling of all an organization’s members.

## 8 Conclusion

Expert finding in an organization is an important task; discovering who knows what can be very challenging, particularly when the organization is large. In such an environment there will likely be many members that an expert finding algorithm has little or no information about, seriously limiting its effectiveness. To tackle this problem, we developed an algorithm that utilizes the organizational hierarchy and propagates expertise scores among neighbors. In our initial investigations we found that neighbors in an organization tend to have similar expertise. This means that they can serve as a reasonable proxy for those with no profiles, and assist in the ranking of employees for whom we have little information. We tested our algorithm with human-generated expert-rating data, and experimented with an automatic evaluation methodology. In both cases, the results showed that adding hierarchical information to a state-of-the-art expert finding algorithm improves retrieval performance.

## References

1. Ackerman, M.S., Wulf, V. and Pipek, V.: *Sharing Expertise: Beyond Knowledge Management*. MIT press, (2002).
2. Adamic, L. and Adar, E.: How to search a social network. *Social Networks*, 27(3):187-203, (2005).
3. Balog, K., Azzopardi L. and De. Rijke, M.: Formal models for expert finding in enterprise corpora. In: *Proc. ACM SIGIR*, pp. 43-50, (2006).
4. Campbell, C.S., Maglio, P.P., Cozzi, A. and Dom, B.: Expertise identification using email communications. In: *Proc. ACM CIKM*, pp. 528-531, (2003).
5. Dom, B., Eiron, I., Cozzi, A. and Zhang, Y.: Graph-based ranking algorithms for e-mail expertise analysis. In: *Proc. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 42-48, (2003).
6. Diaz, F.: Regularizing ad hoc retrieval scores. In: *Proc. CIKM*, pp. 672-679, (2005).
7. Foner, L.N.: Yenta: a multi-agent, referral-based matchmaking system. In: *Proc. of 1st International Conference on Autonomous Agents*, pp. 301-307, (1997).

8. Fang, H. and Zhai, C.: Probabilistic models for expert finding. In: *Proc. ECIR*, pp. 418-430, (2007).
9. Huhns, M., Mukhopadhyay, U., Stephens, M. and Bonnell, R.: DAI for document retrieval: The MINDS project. In *Distributed Artificial Intelligence*, pp. 249-283. London (1987).
10. Hiemstra, D. and Kraaij, W.: Twenty-one at trec-7: Ad-hoc and cross-language track. In: *Proc. 7th Text Retrieval Conference*, pp. 227-238, (1999).
11. Krovetz, R.: Viewing morphology as an inference process. In: *Proc. ACM SIGIR*, pp. 191-202, (1993).
12. Krulwich, B. and Burkey, C.: ContactFinder agent: answering bulletin board questions with referrals. In *Proc. National Conference on Artificial Intelligence*, pp. 10-15, (1996).
13. Kautz, H., Selman, B. and Shah, M.: Referral Web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3): 63-65, (1997).
14. Kleinberg, J.M.: Hubs, authorities, and communities. In: *ACM Computing Surveys*, 31(4es), (1999).
15. Kurland, O. and Lee, L.: Corpus structure, language models, and ad hoc information retrieval. In: *Proc. ACM SIGIR*, pp. 194-201, (2004).
16. Liu, X. and Croft, W.: Cluster-based retrieval using language models. In: *Proc. ACM SIGIR*, pp. 186-193, (2004).
17. Mei, Q., Zhang, D. and Zhai, C.: General optimization framework for smoothing language models on graph structures, In: *Proc. ACM SIGIR*, pp. 611-618, (2008).
18. Miller, D., Leek, T. and Schwartz, R.: A hidden markov model information retrieval system. In: *Proc. ACM SIGIR*, pp. 214-221, (1999).
19. Page, G.E. and Mueller, A.L.: Recognition and utilization of expertise in problem-solving groups: Expert characteristics and behavior. *Group Dynamics: Theory, Research and Practice*, 1(4): 324-328, (1997).
20. Page, L., Brin, S., Motwani, R. and Winograd, T.: The pagerank citation ranking: Bringing order to the web. *Stanford University Technical Report*, (1998).
21. Petkova, D. and Croft, W.: Hierarchical language models for expert finding in enterprise corpora. In: *Proc. of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pp. 599-608, (2006).
22. Ponte, J. and Croft, W.: A language modeling approach to information retrieval. In: *Proc. ACM SIGIR*, pp. 275-281, (1998).
23. Schwartz, M. and Wood, D.: Discovering shared interests using graph analysis, In: *Communications of the ACM*, 36(8): 78-89, (1993).
24. Serdyukov, P. and Hiemstra, D.: Modeling documents as mixtures of persons for expert finding. In: *Proc. ECIR*, pp. 309-320, (2008).
25. Shakery, A. and Zhai, C.: Smoothing document language models with probabilistic term count propagation, *Information Retrieval*, 11(2): 139-164, (2008).
26. Sihm, W. and Heeren, F.: XPertfinder-expert finding within specified subject areas through analysis of e-mail communication. In: *Proc. Euromedia Valencia*, pp. 279-28, (2001).
27. Steeter, L. and Lochbaum, K.: Who knows: A system based on automatic representation of semantic structure. In: *Proc. RIAO*, pp. 380-388, (1988).
28. Tao, Q., Liu, T., Xhang, X., Chen, Z. and Ma, W.: A study of relevance propagation for web search. In: *Proc. ACM SIGIR*, pp. 408-415, (2005).
29. Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2): 179-214.
30. Zhang, J., Ackerman, M.S. and Adamic, L.: CommunityNetSimulator: Using simulation to study online community network formation and implications. In *Proc. of Communities and Technologies*, pp. 28-38, (2007).
31. Zhang J., Ackerman M. and Adamic L.: Expertise networks in online communities: Structure and algorithms, In *Proc. WWW*, pp. 221-230, (2007).