Expert finding in question-and-answer web services

Luís Brandão
6110169
l.silva1@student.uva.nl

Patrick Diviacco
5781205
patrick.diviacco@gmail.com

ABSTRACT

In 2005, the launch of the Expert Finding task in the Enterprise Track of the Text REtrieval Conference (TREC) generated a lot of interest in expertise retrieval, with rapid progress being made in terms of modeling, algorithms, and evaluation aspects. Since then, a large part of the work developed on the fields of expert finding and profiling has been validated experimentally using the W3C collection [7] released by for this task. While TREC boosted the development of the field and provided a common platform for assessment of methods and techniques, using exclusively TREC data to evaluate Experts Finding models does not guarantee that these models generalize well in alternative scenarios.

In this paper we address this issue by evaluating how two expert finding methods, built upon statistical language models and mainly developed under the W3C data set, perform in a public available data set of a question-and-answer (QA) web community.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H3.4 [**Information Systems Application**]: H.4.2 Types of Systems; H.4.m Miscellaneous

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

expert finding, social media, user-centered information retrieval

1. INTRODUCTION

Expert finding addresses the task of finding persons with relevant knowledge or experience about a given topic [24]. In collaboration web services where knowledge exchange is the main activity, one of the greatest challenges is to have a high number of users that actively participate. QA communities are one type of these services where users go to ask and answer questions regarding a specific subject. Since most QA web services are maintained by a community, their success depends on the users engagement and expertise. Expert Finding methods can be used to enhance the user experience by providing mechanisms that enable the presentation of personalized suggestions to the users based on their knowledge and interests. In addition, it can be used to measure the quality of the content by identifying the degree of expertise of the users and define degrees of authority for the users based on their expertise.

Stack Overflow [3] is a QA web community that allows users to ask and answer questions about computer programming languages. The platform combines features of Wikis, Blogs and Forums, and aims to provide free knowledge sharing between software developers worldwide. Running since 2008, the owners estimate that almost 30 % of the professional programmers community - estimated at about 9 million people - have visited the Web site either as a permanent member, or as a visitor [2]. Every month a data dump is publicly released with information about users, tags, badges, posts, comments and votes. The structure of the data and the frequent release of updated dumps opens many possibilities for future expert finding experiments. The different types of users-documents associations and community statistics, for instance, can be used to enhance the above mentioned expert finding methods. Furthermore, the data dumps have additional characteristics such as taxonomy vocabularies that help the users to categorize their content and non-textual features (i.e. voting questions and answers, accepting a submitted answer as question solution), providing useful meta-information about users and documents.

While some work has been developed on Expert Finding in QA web communities ([19], [14], [13]), to the best of our knowledge Stack Overflow was never used for this task. In this paper we explore the Stack Overflow data dump by evaluating the Expert Finding models 1 and 2 proposed by Balog et al. [9], [10] in the April 2010 data dump. We evaluate how Expert Finding methods built upon statistical language models and mainly developed under the W3C data set in this data set.

This paper is structured as follows. In Section 2, we outline relevant research in the field of Expert Finding and how it has been recently applied to QA communities. Section 3 introduces the research questions of this paper. In Section 4, we describe the document collection. In Section 5, we introduce the experimental setup used in this work. In Section 6, we present and discuss the results. Finally, in Section 7 and 8, we conclude and discuss future extensions of this work.

2. RELATED WORK

In this section we start by introducing relevant research conducted in the field of Expert Finding. We finish by presenting research works where these methods were successfully applied to QA web communities.

2.1 Expert Finding

Expertise search and management was initially developed in the field of knowledge management to address the need of identifying experts in corporate environments. The main motivations for expert seeking are the need to find experts to perform a certain task or to get information about a topic [22].

The problem of expert finding was first addressed by creating knowledge databases maintained by users that self-assessed their knowledge. Such approach required a considerable effort from the users that were obliged to frequently fill databases with information regarding their expertise. The data in this kind of systems was easily outdated and hardly captured all the information about the users.

Expert finding systems evolved towards the automation of expert finding by automatically mining corporate internal information. Most approaches of this kind were built specifically for one type of repository, such as emails [13]. This proved to be limited and heterogenous systems that combine documents from different data sources were developed. By 2005 the Text REtrieval Conference (TREC) launched an expert finding task, providing a common platform to assess methods for expert finding [5]. The launch of this task was followed by an increasing of the interest of the information retrieval community in the field.

The current state of the art on expert finding comes from the work by Balog, de Rijke and Azzopardi [9], [10]. The two models proposed are based on probabilistic language modeling techniques that are commonly used in Information Retrieval tasks. In the abstract, their approach is grounded on the estimation of how likely a query could have been generated by each candidate. To do so, they propose distinct ways of determining the probability of a candidate ca given a query q:

$$p(ca|q) \propto p(q|ca) \cdot p(ca) \tag{1}$$

They present four models, divided in two groups, that mainly differ in the way the probability of query given a candidate, $p(q|ca)$, is estimated. The first set of models, referred as Candidate Models (Models 1 and 1B), are generated by first collecting all the documents associated with a candidate and then determining the topics of such documents. The second set of models, called Document Models (Models 2 and 2B), first identifies the documents that belong to a given topic and then determine who are the users associated with these documents.

The difference between the Models 1 and 2 and their homonyms with the suffix B, which we will call B models from now on forward, is on the Method used to calculate the probability of a term. In Models 1 and 2, only the document determines the probability of a term; in Models 1B and 2B the probability of a term is determined by the documents and the candidates. In the B models, all the references to a candidate are identified in the documents and the words on the left or on the right of each occurrence of the candidate in a certain distance window are considered areas of expertise of a candidate. On the other hand, the models 1 and 2 assume that only the documents produced by the candidate are relevant.

Candidate Models can be easily deployed, because they do not need a dedicated index, and outperform the Document Models in most of the cases. This is probably due to the fact that they estimate the relevance of the text content related to a person on the much lower and hence less ambiguous level. Nevertheless, [12] and [21] show that under certain circumstances the Document Models can outperform the Candidate Models and that a combination of both models results in substantial improvements. The use of external information, collected from the web, was also shown to improve the results as indicated before on previous experiments [17], [20].

Document-candidate associations form an essential part of the candidate and document models. Associations define the probability that a document d is associated with candidate ca, and are based on named entity recognition: in [9], it has been demonstrated that "Full Name match" of the candidates gives better performances than "Last Name match" or "Email match". Interestingly, for email corpora, it was found that being included in the CC field on an email was more important than being the author of an email, for use as expertise evidence [11].

2.2 Expert Finding in QA communities

While the methods described in the previous section can be applied to QA services, this kind of communities usually have a structure and an interaction model between the users that gives rise to new challenges. In this section we discuss relevant research in the field applied to QA communities.

Content selection When language models are used to model and identify experts, different data sub-sets can be used. Liu, Croft and Koll identified four possible different content selections for expert modeling based on content analysis of QA communities [19]. The following content sets are considered in their experiment setup:

- all previously answered questions by a user, both question and answer texts (All QA pairs)
- all previously answered questions, question texts only (All Qs)
- one of the previously answered questions, both question and answer texts (i.e. "Single QA pair")

- one of the previously answered questions, question text only (i.e. “Single Q”).

These models were evaluated in a pool of 852316 QA pairs from a question and answer web service. Five data sets were created from these QA pairs, each having a different minimum number of question answered by each user. The experiments results don’t differ significatively between the four data sub-sets, but the one that considers all previously answered questions with question texts only gives the most consistent results across the different data sets.

Retrieving relevant documents In document-centric approaches, first the relevant documents given a specific topic are retrieved and then associated candidates are considered. For this reason, the quality of the retrieved documents strongly influence the system performance. Approaches to measure the quality of answers in QA communities can be classified into three types [14]:

- Approaches based on Information Retrieval and Natural Language Processing techniques. Language models can be generated for submitted questions, answers, user profiles, and provided tags.
- Approaches based on statistical analysis use non-textual features [16] such as answerer’s acceptance ratio (the ratio of best answers to all the answers that the answerer answered previously), the score assigned to a question or answer by community users, the answerer activity over time and textual features [8] such as the answer length, number of words in the answer with a corpus frequency larger than a certain number or the non stop-word overlap between question and answer, etc.
- Approaches based on link analysis, such as HITS and PageRank, that consider the network structure of web communities, to measure users reputation and rank users and answers.

Using link analysis Campbell et al. [13] proposed the use of HITS algorithm which performed better than statistical approaches such as ranking the candidates according to their ”in-degree” (number of answered questions). HITS assigns two scores to each user: a hub score and an authority score. A good hub is a user who is helped by many expert users, and a good authority (an expert) is a user who helps many good hubs. While replying to many questions implies that one has high expertise, asking a lot of questions is usually an indicator that one lacks expertise on some topics.

Zhang et al, address the expert finding problem based on the users network structure [23]. Their work, revisits the work by Campbell et al. [13] seeking to overcome the limitations of this study regarding the size of the networks used. They use graph-based algorithms on the social network of a Java Forum and analyse how the structure of the network affects the performance of algorithms like HITS or PageRank. The results show that this method can be used for expert finding on question answer websites.

Jurczyk [18] proposes using a PageRank-like algorithm to generate a measure that not only considers how many other people one helped, but also whom he/she helped (expertise rank should be boosted not just because they were able to answer a question, but because they were able to answer a question of someone who herself had some expertise). Similarly, Hong [15] discussed the possibility of using a PageRank-like scheme for unsupervised topical link analysis for user reputation models.

3. RESEARCH QUESTIONS

Given the interesting nature of QA communities, and the importance of organizing their content and the community expertise, the purpose of this paper is to evaluate how Expert Finding methods based on statistical language models perform in such environments and can enhance their functionalities. More specifically, this paper focus on the following research questions:

- How do state-of-the-art algorithms developed on the W3C data set perform in the alternative scenarios such as QA communities of the type described above?
- More specifically, does the document-based model outperforms the candidate-based model in a QA system environment?
- Is taxonomy enhancing the expert finding models? Can taxonomy terms associated with the documents be included in the documents language models ?
- Can non-textual features (deriving from community statistics, such as ”best / accepted answer”) be used to improve the experts lists ranking ?

4. DOCUMENT COLLECTION

4.1 Dataset

We use the April 2010 data dump provided by Stack Overflow [4]. The document collection contains all the questions and answers posted on the web site between July 31, 2008 and March 31, 2010. In addition, contains one or more tags per question that identify its topic(s). Questions considered solved are the ones marked by the question owner as having an Accepted Answer. All posts, both questions and answers, are scored by the users as follows:

Table 1: Data collection posts scoring system

Event	Score
user’s answer is accepted	+15
user’s answer is voted up	+10
user’s question is voted up	+5
user’s question or answer is voted down	-2

We use a subset of the document collection, obtained by filtering the document collection to select the 50000 questions (plus answers) with the highest score of the 10 tags with more questions. Only questions with one Accepted Answer (marked as solved) were chosen. The stop words and code snippets were removed from the collection. Table 2 contains the number of posts per selected tag.

Table 2: Questions and Answers per tag

Tag	Questions	Answers
c-sharp	13888	47565
java	8001	26924
.net	7819	28754
asp.net	3657	12632
php	4100	13710
javascript	4192	13613
c++	6989	24006
jquery	2420	7612
python	5478	17382
iphone	1858	6460
Total	50000	231388

4.2 Content selection and non-textual features

As mentioned in section 2.2, when language models are used to model and identify experts in QA communities, different data subsets can be used. Non-textual features in QA communities are meta-information associated with the documents, based on community statistics. These features, such as questions and answers votes, number of views, best / accepted answer (solution), can be used as quality criteria to select a subsets of the data. We use them to define four subsets of the document collection for our experiments:

- all previously answered questions - question and accepted answer text
- all previously answered questions - question, accepted answer and associated tags text
- all previously answered questions - question and all answers text
- all previously answered questions - question, all answers and associated tags text

4.3 Taxonomies

QA communities questions are annotated with community-edited tags to better organize the content and help users to categorize question and related answers. Tags associated with questions can be integrated into documents and candidate language models, in order to better define the expertise areas. Addressing taxonomies, may be necessary if QA services provide them, since users might define the context of their questions with tags (i.e. specifying the programming language with the tag "Java") and not provide redundant information into the question text.

In order to evaluate if taxonomies enhance expert finding models, two different types of queries are used in the experiment:

- queries including questions content
- queries including questions content and associated tags.

4.4 Document-candidate Associations

Document-candidate associations form an essential part of the candidate and document models. Associations define the probability that a document d is associated with candidate ca, and, in common expertise corpora, are based on named entity recognition. There are several types of associations in QA communities: a document can be associate to the user submitting the question, the answerers or a non-textual feature such as user votes ("Thumb up / Thumb down"). These associations have different strength, for example an user submitting the best answer to the question is intuitively more strongly related to the document than an user voting the question or one of the answers. Only strong user-document associations have been used in the experiments: an user is associated to a document only if he is the answerer of the selected question.

5. EXPERIMENTAL EVALUATION

In this section we address the questions of the previous section by presenting a set of baseline approaches, based on generative language modeling, aimed at finding experts for communities questions and expertise areas.

The expertise corpora for the experiment consists of a QA community database. Briefly, in QA corpora, a document is composed by a question together with its answers. Users can contribute by both submitting the question or the answers. Only one answer can be accepted by the questioner as question solution. A topic expert is an user who answered correctly many questions (submitted many accepted answers) annotated by community with that specific topic.

An Expert Finding system gets a query with some content related to specific topics as input and produces a list of experts for that topics as output. In this experiment, a list of community questions (queries) are submitted in each test and lists of experts related to the input questions are returned. The system is evaluated by verifying if the users who effectively provided the best answer to the questions are included in the predicted experts lists.

5.1 Expert Finding models

We use the candidate and document models proposed by Balog et al. [9], [10]. We evaluate and compare them under different settings, over the Stack Overflow document collection. In [9], Balog et al. demonstrated that the document model outperforms the candidate model on the W3C collection from the Enterprise Track. However, [12] and [21] show that under certain circumstances the Document Models can outperform the Candidate Models and that a combination of both models results in substantial improvements. For this reason, the quality of the retrieved documents strongly influence the system performance.

5.2 Experiment Configuration

Software We use the Entity and Association Retrieval System (EARS) [1] in the experiments. EARS is an open source toolkit for entity-oriented search and discovery in large text collections. EARS supports two main tasks: finding entities ("Which entities are associated with topic X?") and profiling entities ("What topics is an entity associated with?"). It implements the Candidate and Document Models described in section 2.1. EARS is built on top of the Lemur Toolkit [26], an open-source toolkit designed to facilitate research in language modeling and information retrieval. The results are evaluated using the evaluation module trec_eval [6].

Queries We test both candidate and document models by querying the system with 50000 questions selected from the dataset. For each query, a list of 10 predicted experts is returned by the system.

Ground truth The user who provided an accepted answer for a specific question is considered the unique expert for the question. Each evaluation is measured by computing the Mean reciprocal rank (MRR) for each question expert over the top 10 experts predicted by the system.

Figure 1: Experiment description

Experiment	Expert Finding Model	Query	Content Selection (Non textual Features
1.1	Model 1 (Candidate Model)	Questions	question + accepted answer
1.2			question + all answers
1.3		Questions + Tags	question + accepted answer
1.4			question + all answers
2.1	Model 2 (Document Model)	Questions	question + accepted answer
2.2			question + all answers
2.3		Questions + Tags	question + accepted answer
2.4			question + all answers

6. RESULTS AND DISCUSSION

Figure 2 summarizes the experiment results.

Figure 2: Results measured with Mean Average Precision

Expert Finding Model	Content Selection (Non textual Features	Queries	
		Questions	Questions + tags
Model 1 (Candidate Model)	question + accepted answer	**0.86**	**0.87**
	question + all answers	0.80	0.81
Model 2 (Document Model)	question + accepted answer	0.57	0.57
	question + all answers	0.49	0.50

Models comparison A quick scan of Figure 2 reveals that the candidate model significantly outperforms the document model over all settings. This result contrasts with the models performance over the W3C collection, and it might be explained by the different nature of both collections. In Stack Overflow and, in general, in all QA systems, only 2 % of the users, answer most of the questions on specific topics. Consequently, each expert is associated with a high number of documents. Furthermore, documents in QA corpora are usually short. Questions and answers consist of few sentences, and are considerably shorter than publications, reviews, or other kind of documents (i.e. W3C expert finding collection). Language models based on short documents without redundant content are not very descriptive. By opposition, candidate language models based on considerable amounts of documents describe extremely well the candidates knowledge areas.

In Model 2 (document model), the quality of the retrieved documents used to find the associated experts is crucial. If the documents are short and they don't have optimal language models the method has low performance. For the opposite reason, candidate models based on high quantities of documents associated to the candidate, perform better.

Since most QA systems have this characteristics in common, we believe that candidate models are more appropriate for QA communities.

Content selection and non-textual features Filtering the dataset with non-textual features and removing the non-accepted answers from the collection enhances the expert finding models (MRR is ≈ 0.6 higher). Although removing non-accepted answers makes the documents shorter, and, intuitively, makes the document language models performing worse, the global performance of the methods improves under this setting. Such improvement might be explained by considering which part of the documents are filtered. Differently from the W3C corpora, where the contribute of each user to the document is not easily individual, in QA corpora each answer is associated to a specific user, and thus his contribute is well-defined. Thus, removing non-accepted answers from a document, implies removing those sections that are not related to the user who correctly answered the answer (the expert). As a consequence, this approach refine document language models and optimize the terms weights.

Taxonomies Addressing taxonomies, by including associated tags into queries and documents, didn't significantly improve the methods performance. Queries including questions content and associated tags, performed slightly better (MRR is ≈ 0.01 higher) than queries including only questions content. We believe that such behavior is due to two main reasons:

- The documents could already include the taxonomy terms, since users write redundant content and thus no new terms are added to the language models.
- It might also be that including associated tags (representing the document topics) in the language models is not fundamental to categorize the document since terms strongly related with the topic are already present in the document. For example, including the tag "Java" in a document already containing Java-related terms such as "abstract class", "extends" and "virtual machine", might not improve the language model of the document significantly.

7. CONCLUSION

In this paper state-of-the-art expert finding methods have been applied to QA communities corpora in order to evaluate how they performed in this new environment.

Furthermore, the methods have been evaluated with different settings, exploiting QA communities specific characteristics, such as non-textual features and taxonomy, in order to see if under certain circumstances their performance is enhanced.

The experiment reveals that the candidate model significantly outperforms the document model over all settings. This results contrast with the models performance over the W3C collection, and it might be explained by the different nature of both collections. In QA corpora, each expert is associated with a high number of documents, and documents are usually short. Consequently, document models performs worse and candidate models performs better.

Filtering the dataset with non-textual features and removing the non-accepted answers from the collection enhances the expert finding models (MRR is ≈ 0.6 higher). Removing non-accepted answers from a document, implies removing those sections that are not related to the user who correctly answered the answer (the expert). As a consequence, this approach refine document language models and optimize the terms weights.

Finally, addressing taxonomies, by including associated tags into queries and documents, didn't significantly improve the methods performance.

In the next section, a list of future improvements for expert finding methods in QA communities is provided. Future work is discussed for each experiment aspect.

8. FURTHER WORK

In this section a list of future improvements for expert finding methods in QA communities is provided. Future work is discussed inherently to content selection, document-candidate associations, taxonomy, timestamps and link analysis.

Content selection The "non textual-features" associated with each document, in particular questions and answers scores, and number of views for each document help users to determine the quality and usefulness of the documents. It would be worth to evaluate the models performance over a filtered dataset (a subset consisting of the documents with the highest scores). The QA community document scores based on users votes ("Thumb up / Thumb down") could be refined with additional statistics such as the number of views or the reputation of questioners and answerers. Users reputation could be computed with their previous posts scores (both questions and answers) and new documents submitted by users with high/low reputation could be have higher/lower ranking.

Document-candidate associations A document is associated to a user who submitted the question or one of its answers. A possible alternative would incorporate secondary activities, like questions / answers votes and comments as an evidence of expertise. Different kind of associations could have different strength values, expressing how much a user is related to the document.

Taxonomy Community original taxonomy vocabularies could be improved by clustering similar tags or creating tags hierarchies to better organize the expertise areas of the community. For example, the tag "web development" is often used instead of "html, css, javascript" and tags such as "Adobe Flash" and "Actionscript" are strongly related topics.

Timestamps Timestamps could be useful in order to determine expertise trends in the community over time and to determine which users are currently focusing on specific topics. For example, an expert could not be anymore updated on a specific topic, since its interests changed over time.

Link analysis Another research path that we believe being worth exploring is the application of generative language models techniques combined link analysis techniques to QA communities. Language-based expert models such as the document-model proposed by Balog et al. [9], or models based on combination of candidate-centric and document-centric approaches [12] could be enhanced considering the social network structure of QA services, in which questioners and answerers are semantically (via expertise areas) connected by the documents (in this case, question - answers). The results of such models could be improved with information provided by link analysis approaches such as HITS or Pagerank, as proposed in [23], [18]. For instance, expert lists and retrieved documents could be re-ranked according to questioner and answerers reputation. In addition, terms of generated language models could be weighted differently according to the prestige of the source document.

9. REFERENCES

[1] Entity and association retrieval system (ears). `http://code.google.com/p/ears/`.

[2] Learning from stackoverflow.com [video]. `http://www.youtube.com/watch?v=NWHfYlvKIQ`.

[3] Stack overflow. `http://stackoverflow.com`.

[4] Stack overflow data dump. `http://blog.stackoverflow.com/2009/06/stack-overflow-creative-commons-data-dump/`.

[5] Trec (2005) enterprise track. `http://www.ins.cwi.nl/projects/trec-ent/wiki/`.

[6] trec eval. `http://trec.nist.gov/trec_eval/`.

[7] W3c collection (trec). `http://research.microsoft.com/en-us/um/people/nickcr/w3c-summary.html`.

[8] E. Agichtein, C. Castillo, D. Donato, and A. Gionis. Finding high-quality content in social media. *Proceedings of the international conference on Web search and web data mining*, Jan 2008.

[9] K. Balog, L. Azzopardi, and M. D. Rijke. Formal models for expert finding in enterprise corpora. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, Jan 2006.

[10] K. Balog, L. Azzopardi, and M. D. Rijke. A language modeling framework for expert finding. *Information Processing and Management: an International Journal*, Jan 2009. Notes Main paper.

[11] K. Balog and M. Rijke. Finding experts and their eetails in e-mail corpora. *Proceedings of the 15th international conference on World Wide Web*, May 2006.

[12] K. Balog and M. D. Rijke. Combining candidate and document models for expert search. *Text REtrieval*

Conference 2008, Jan 2008.

[13] C. Campbell, P. Maglio, and A. Cozzi. Expertise identification using email communications. *Proceedings of the twelfth international conference on Information and knowledge management*, Jan 2003.

[14] L. Chen and R. Nayak. Expertise analysis in a question answer portal for author ranking. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, 1:134 – 140, 2008.

[15] L. Hong, Z. Yang, and B. Davison. Incorporating participant reputation in community-driven question answering systems. *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*.

[16] J. Jeon, W. Croft, J. Lee, and S. Park. A framework to predict the quality of answers with non-textual features. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, Jan 2006.

[17] J. Jiang, S. Han, and W. Lu. Expertise retrieval using search engine results. *The SIGIR 2008 workshop on future challenges in expertise retrieval*.

[18] P. Jurczyk and E. Agichtein. Discovering authorities in question answer communities by using link analysis. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, Jan 2007.

[19] X. Liu, W. Croft, and M. Koll. Finding experts in community-based question-answering services. *Proceedings of the 14th ACM international conference on Information and knowledge management*, Jan 2005.

[20] P. Serdyukov and D. Hiemstra. Being omnipresent to be almighty: The importance of the global web evidence for organizational expert finding. *Proceedings of SIGIR 2008 Workshop on Future Challenges in Expertise Retrieval*, Jan 2008.

[21] P. Serdyukov and D. Hiemstra. Modeling documents as mixtures of persons for expert finding. *Lecture Notes in Computer Science*, Jan 2008.

[22] D. Yimam-Seid and A. Kobsa. Expert finding systems for organizations: Problem and domain analysis and the demoir approach. *Journal of Organizational Computing and Electronic Commerce*, Jan 2003.

[23] J. Zhang, M. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. *Proceedings of the 16th international conference on World Wide Web*, Jan 2007.

[24] J. Zhang, J. Tang, and J. Li. Expert finding in a social network. *Lecture Notes in Computer Science*, Jan 2007.