Thread-based Expert Finding

Jangwon Seo W. Bruce Croft
Center for Intelligent Information Retrieval
University of Massachusetts, Amherst
{jangwon, croft}@cs.umass.edu

ABSTRACT

Identifying experts is an important part of search quality in online community sites such as forums. A number of word-based and structure features have been used for the task of expert finding. In this paper, we focus on hierarchical structures in online community sites: posts, threads and thread structures. Through empirical comparisons on two collections, we show that thread structures can be helpful for expert finding.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms, Performance, Experimentation

Keywords

expert finding, thread structure, link analysis

1. INTRODUCTION

Online communities are virtual spaces where people who are interested in a specific topic gather and discuss in depth a variety of sub-topics related to the topic. Although community members can equally discuss with other members, there are members across the expertise spectrum from non-experts to experts. Expert identification involves finding experts on a given topic.

Expert identification in online communities is of importance for the following two reasons. First, online communities can be viewed as knowledge databases where knowledge is accumulated by interactions between the members. That is, we read articles in online communities to get information on specific topics. If we find articles written by experts, we tend to have more confidence in their content. On the other hand, in terms of communication dynamics, online communities are spaces where non-experts can communicate with experts. In the real world, communicating with experts is not only difficult but also expensive. However, we can relatively easily communicate with experts in online communities if we know who they are.

Expert finding in online communities has been explored by many researchers. Campbell et al. [1] employed graph-based ranking algorithms to rank experts in an email network. Zhang et al. [11] reviewed expertise ranking algorithms and performed modeling of social network in an online forum using simulation techniques. Viégas and Smith [10] identified different behavioral patterns of authors visualizing authors' activities in Usenet newsgroups. Since 2005, The TREC community has organized an expert finding task in enterprise environments [9]. Serdyukov et al. [8] introduced relevance propagation modeling through candidate experts nodes and document nodes for expert finding. Jurczyk and Agichtein [3] used a link analysis algorithm to rank authors in community based-QA portals.

Our work is different from previous work in that we explicitly employ thread structures for a graph-based ranking algorithm. A thread is a focused topic-centric discussion unit and is composed of posts created by community members. Our proposition is that structures in threads will be helpful for expert finding. In this paper, we empirically analyze and compare various techniques including a thread-structure based technique.

2. ALGORITHMS

The expert finding task is to locate people who are experts in a given topic. An effective approach to this task is to divide the task to two sub-tasks, i.e. to find a topically relevant document subset and to find experts in the subset [1, 8, 9]. To find the topical subset, we introduce two classes of expertise graph construction methods: post-based and thread-based graph construction. For finding experts in the expertise graph, we suggest a variation of a random walk algorithm which analyzes the graphs to rank experts.

2.1 Post-based Graph Construction

In online communities such as forums, a post is an atomic topical unit used to communicate with community members. A set of relevant posts can be considered as a relevant subset for expert finding in that a post usually address a topic and a post is created by only one person. We assume that we can find experts by analyzing authorship of relevant posts.

To retrieve a set of relevant posts to a given topic, we rank posts by the query likelihood which is estimated as follows:

$$P(Q|D) = \prod_{q \in Q} P(q|D) = \prod_{q \in Q} \left(\frac{tf_{q,D} + \mu \cdot tf_{q,C}/|C|}{\mu + |D|} \right) \quad (1)$$

where D is a post, C is a collection, q is a query term in query Q and μ is a Dirichlet smoothing parameter.

Once we have a ranked list by the query likelihood, we can build a graph using top N posts. First, we make document nodes with the posts. Next, we make candidate expert nodes with unique authors of the posts. Finally, we make directed edges from document nodes to candidate nodes so that each candidate is reachable only from the posts written by the candidate. Figure 1(a) presents a post-based graph example.

SIGIR'09 SSM Workshop, July 23, 2009, Boston, Massachusetts, USA.

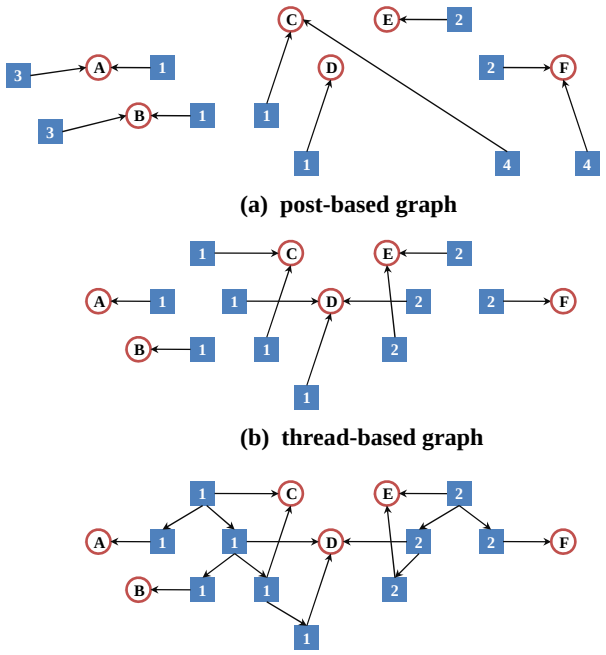

Figure 1: Graphs by different construction methods. A circle is a candidate node and a square is a post node. A number in each square is an ID of a thread that the post belongs to.

As you see, each candidate and its posts make a connected graph.

2.2 Thread-based Graph Construction

A thread is started on some subject by an initiator post and grows as people write posts to discuss the subject. That is, a thread can be considered a group of posts which address a similar topic. Therefore, while a reader can obtain an individual's opinion from a post, one can obtain a group's opinion from a thread. Further, threads often give better understanding about a topic by contexts or conversational flows in the threads. For example, post A says, "What's new in iPod X?" and "Its sound quality is better than the previous models," replies post B. By looking at only post B, we cannot understand the topic. Moreover, if a query is "iPod X", then post B is not even considered as a relevant post in post-based retrieval. Therefore, we here consider a set of relevant threads as a subset for expert finding.

We concatenate all posts in a thread to make a bag-of-word language model for the thread. We then retrieve the top N ranked threads by Equation (1). Then, for all posts in the threads, we build a graph in the same manner as the post-based algorithm. Figure 1(b) shows an example. We can see that the thread-based graph uses a different set of posts from the post-based graph.

Now we go one step further and consider thread structures. A thread structure is composed of reply relations in a thread. In most online communities, many-to-many communication is usual in a thread, and accordingly, readers can be confused with who talks to whom, particularly in long threads. With thread structures, this problem is resolved because reply relations distinguish each context from others. Further, in terms of expert finding, we can expect thread structures to help identifying influential posts. We assume that if a candidate writes an influential post, then one is likely to be an expert. Note that we here assume that a child post replies to only a parent post because most threaded-view systems work under this assumption.

Recently, a number of online communities have supported the threaded-view, where reply relations are explicitly displayed using indentation or tags. We can easily extract thread structures from such communities. Even when a forum does not support the threaded-view, we can try to infer the thread structures using thread structure discovery algorithms recently introduced [5, 7].

Once we have thread structures, we can make post-to-post links with them. However, here is a question to consider. In the post-to-candidate links, the direction of the links from post to candidate looks natural because the authorities are the candidates rather than the posts and a document can be considered as a citation from a candidate's knowledge. In contrast, the direction of post-to-post links is somewhat vague. If a parent-child post pair has a question-answering relation, then the authority is the child. On the other hand, if the pair has a suggestion-agreement relation, then the parent is likely to be authoritative. Even in a collection, there can be various relations. Therefore, we report results for parent-to-child as well as child-to-parent relationships in the experiments.

2.3 Expertise Ranking

For expertise ranking, we use a random walk algorithm similar to the PageRank algorithm [4, 6]. To customize the PageRank algorithm, we make a modification. A random walk matrix of the PageRank is defined as follows:

$$\bar{\bar{\mathbf{P}}} = \alpha\bar{\mathbf{P}} + (1-\alpha)\mathbf{e}\mathbf{e}^T/n \tag{2}$$

where $\mathbf{e}$ is the column vector of all ones, n is the order of the matrix, $\bar{\mathbf{P}}$ is an adjacency matrix where rows of dangling nodes are replaced by $\mathbf{e}^T/n$, and α is a parameter to control the effect of random jumps. The second term $\mathbf{e}\mathbf{e}^T/n$ is a random jump matrix in order to make the random walk matrix irreducible, which is a necessary condition for convergence of the PageRank vector.

We modify this random jump term. First, we prohibit random jumps between heterogeneous nodes, i.e. post-to-candidate or candidate-to-post. When considering a random surfer, jumps between documents sound natural. Further, jumps between candidates can be understood as communication outside the forum. However, post-to-candidate can be considered as somewhat weird behaviors such as random authorship. We would like to avoid these jumps. Second, when reading a post, a random suffer is likely to read other posts in the same thread because a user view usually displays multiple posts in a thread. That is, the probability of jump to posts in the same thread is possibly higher than that of jump to any other posts. Therefore, we consider a new random jump matrix as follows:

$$\mathbf{E}_{ij} = \begin{cases} 1/|V_C| & \text{if } i,j \in V_C \\ \beta/|V_{T_k}| + (1-\beta)/|V_D| & \text{if } i,j \in V_{T_K}, \exists k \\ (1-\beta)/|V_D| & \text{if } i,j \in V_D \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where V_C is a set of candidate nodes, V_T is a set of post

nodes in any thread, V_D is a set of post nodes, and β is a parameter. This matrix is used for Equation (2) instead of $\mathbf{e}\mathbf{e}^T/n$. The final random walk matrix is stochastic and irreducible because nodes are fully reachable between candidates or posts, each post is reachable from candidates by substitutions for dangling nodes, and a candidate has at least one incoming edge. Therefore, this matrix guarantees a convergence of the PageRank vector. Both parameters α and β are set to 0.85 which is known as a magic number in the PageRank studies [4].

3. EXPERIMENTS

We conduct experiments on two different types of collections: an email archive and a forum.

3.1 Email Archive

Email archives or newsgroups are old-style online communities but are still active in technical areas.

As an email archive collection, we used the 'lists' subcollection of the W3C collection for TREC enterprise track [9]. The collection was crawled from the mailing list[1] of the World Wide Web Consortium (W3C).The collection contains 72,214 threads and each thread includes 2.1 messages on average.

Since the W3C collection has been used for the expert finding task of the TREC enterprise track 2005 and 2006, there is a relevance judgment set provided by TREC. Since topics for TREC 2005 were used for the pilot evaluation and there is no manual judgment for them, we used only topics for TREC 2006, which contains 49 queries and 8,351 relevance judgments.

3.1.1 Thread Structure Recovery

The W3C collection provides thread structures in the '`thread.html`' file in each group archive. However, many of these thread structures are wrong. We frequently find cases where an earlier email replies to a later email. This is because the 'msg-id' and 'inreply-to' tags in email headers are often lost. A thread of emails is usually constructed by matching tags. If they are missing, then email archive tools infer threads using heuristics such as title matching. Such inferences are often inaccurate.

To obtain more accurate thread structures, we train a classifier. To build an annotation set for thread structure recovery training, we refined the thread structures by picking threads only composed of emails whose 'inreply-to' tag matches a 'msg-id' tag of any other post in the same thread. Finally, we obtained 1,635 threads which contain at least 3 emails.

We extracted various pairwise features from each pair in the annotated threads. See Seo et al. [7] for the detailed description of these features.

Since a thread structure is constructed by reply relations, this task can be interpreted as finding the most likely parent message given a child post. That is, this can be considered a ranking task, considering the parent message and the child message as a relevant document and a query, respectively. To address this task with the heterogeneous features, we used the ranking SVM technique [2]. 1,535 threads of the annotated set were used as training data and the remaining 100 threads were used as test data. Through this procedure, we could learn a highly accurate classifier. The recall score of reply relations on test data, i.e. how many reply relations are correctly identified, was 0.9617. Finally, this learned classifier was applied to all other threads in the collection than the annotation set for thread structure recovery.

[1] http://lists.w3c.org/

	MAP	P@5
Post	0.2607	0.5306
Thread	0.2759^{α}	0.5429
Thread Structure (c→p)	$0.2778^{\alpha\beta}$	$0.5592^{\alpha\beta}$
Thread Structure (p→c)	0.2757^{α}	$0.5592^{\alpha\beta}$

Table 1: Expert finding results for different graph construction methods on the W3C collection. 'Post', 'Thread' and 'Thread Structure' represent the post-based, thread-based, and thread structure-based graph construction methods, respectively. (c→p) and (p→c) mean the direction of child-to-parent and parent-to-child for post-to-post edges. Superscripts α and β indicate statistically significant improvements on 'Post' and 'Thread', respectively. (the paired randomization test with p-value < 0.1)

3.1.2 Graph Construction

To build a post-based graph, we retrieved top 1,000 posts for each topic using Equation (1), where the Dirichlet smoothing parameter was set to 450 that is the average length of a post. Authorship information was extracted from 'From' field of each message. Using these posts and author information, a post-based graph for each topic was constructed.

Note that we did not use the 'To' or 'Cc' fields to extract authors. Since the W3C collection was collected from an email archive, such fields exist. However, generally, most online communities provide only author information and posts are broadcast to all community members. To simulate this situation, we consider only authors in graph construction.

The same process was employed to build a thread-based graph. The differences are that the Dirichlet smoothing parameter was set to 1000 that is the average length of a thread and top 500 threads were retrieved for each topic because 500 threads include the similar number of authors to that do 1,000 posts, i.e. approximately 2,000 authors. For thread structure-based graph, the reply relations inferred by thread structure recovery were used.

3.1.3 Results and Discussion

Results of expertise ranking are reported using two metrics: Mean Average Precision (MAP) and precision at top 5 (P@5). We considered a judged document whose relevance grade is equal to 2 as relevant. Table 1 presents the results.

All the thread-based methods show better performance than the post-based method. Particularly, thread-structure based methods outperform the post-based method. Further, the thread structure-based technique using the direction of child-to-parent is significantly better than the thread-based method. The change of performance depending on the direction of post-to-post edges is not noticeable.

3.2 Forum

Online forums are currently one of the most popular online communities. We can easily find online forums covering many topics ranging from very casual to professional in nature.

Building test collections for expert finding is known to be very expensive even compared to building test collections for ad-hoc retrieval. This is because annotators should judge relevance by reading a number of documents written by an author or should be members of the community so that they can easily recognize the experts. To avoid this difficulty, we used automated test collection generation.

The Apple Discussions[2] provides separate forums for each product by Apple, Inc. Since these forums are divided by fine-grained categories, we can assume that each forum addresses a topic. That is, we consider an individual forum as a topically relevant thread set. We chose 30 forums so that the topics are as disjoint as possible. Table 2 shows examples of the chosen forums. From each forum, we crawled 30 randomly selected pages. Since each page contains 15 threads, we obtained 450 threads in total. Further, each forum of the Apple Discussions provides a top 10 user list based on points which are calculated by the number of replies and the quality of user feedback. We used this list as the gold standard for evaluation.

3.2.1 Thread Structure and Graph Construction

Forums in the Apple Discussions support the threaded-view, that is, the user view displays the reply relation. Since this information is embedded in HTML tags, we can easily extract the reply relation.

Given that crawled forums are relevant thread sets, we can construct only thread-based graphs. Therefore, in this section, we do not compare thread-based methods to post-based methods. Rather, we investigate effectiveness of different thread-based methods. Therefore, we constructed a thread-based graph for each topic (or forum), and we used the extracted reply relations for a thread structure-based graph.

3.2.2 Results and Discussion

Since we have only the top 10 users for each forum, it is not reasonable to treat all users behind top 10 as novices. Therefore, we use recall-based metrics rather than precision-based metrics to observe how well the top 10 users are identified. We report recall scores at 10, 20 and 50 (R@10, R@20 and R@50). Table 3 shows the results.

The thread structure-based method using the direction of parent-to-child for post-to-post links outperforms the thread-based method. On the other hand, using the direction child-to-parent, the thread structure hurts performance. This shows that the Apple forums are considerably biased toward the post relations where replies usually have the authorities, e.g., question-answering relations. Therefore, depending on the characteristics of online communities, the choice of the direction of links between posts can be critical.

4. CONCLUSION

We introduced expertise graph construction methods and a variation of a random walk algorithm for expert finding. Using two different online community collections, we empirically showed that thread structures can be helpful for expert identification. In particular, to appropriately define relations between nodes according to collections can be critical. This work is just preliminary work. We plan to investigate various networks between experts and documents further and study more effective expert finding techniques in online communities.

[2]http://discussions.apple.com/

Product Category		Forum Title
iPod shuffle	>	Using iPod shuffle (Second Generation)
iWork '09	>	Keynote '09
Safari	>	Safari for Mac

Table 2: Examples of the Apple Discussion forums used for the test collection

	R@10	R@20	R50
Thread	0.6667^{β}	0.8367^{β}	0.9500^{β}
Thread Structure (c→p)	0.6500	0.8167	0.9300
Thread Structure (p→c)	$0.6933^{\alpha\beta}$	$0.8600^{\alpha\beta}$	$0.9633^{\alpha\beta}$

Table 3: Expert finding results for different graph construction methods on the Apple forums. Superscripts α and β indicate statistically significant improvements on 'Thread' and 'Thread Structure (c→p)', respectively. (the paired randomization test with p-value < 0.05)

5. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval (CIIR) and in part by NSF grant #IIS-0711348. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

6. REFERENCES

[1] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In *CIKM '03*, 2003.
[2] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, 2002.
[3] P. Jurczyk and E. Agichtein. HITS on question answer portals: exploration of link analysis for author ranking. In *SIGIR '07*, 2007.
[4] A. N. Langville and C. D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3), 2003.
[5] C. Lin, J.-M. Yang, R. Cai, X.-J. Wang, W. Wang, and L. Zhang. Modeling semantics and structure of discussion threads. In *WWW '09*, 2009.
[6] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.
[7] J. Seo, W. B. Croft, and D. A. Smith. Online community search using thread structure. In *CIKM '09*, 2009.
[8] P. Serdyukov, H. Rode, and D. Hiemstra. Modeling multi-step relevance propagation for expert finding. In *CIKM '08*, 2008.
[9] I. Soboroff, A. P. de Vries, and N. Craswell. Overview of the TREC 2006 enterprise track. In *TREC 2006*.
[10] F. B. Viégas and M. A. Smith. Newsgroup Crowds and AuthorLines: Visualizing the activity of individuals in conversational cyberspaces. In *HICSS-37*, 2004.
[11] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *WWW '07*, 2007.