

Online Community Search Using Thread Structure*

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

ABSTRACT

Online communities are valuable information sources where knowledge is accumulated by interactions between people. Search services provided by online community sites such as forums are often, however, quite poor. To address this, we investigate retrieval techniques that exploit the hierarchical thread structures in community sites. Since these structures are sometimes not explicit or accurately annotated, we use structure discovery techniques. We then make use of thread structures in retrieval experiments. Our results show that using thread structures that have been accurately annotated can lead to significant improvements in retrieval performance compared to strong baselines.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Retrieval Models

General Terms

Algorithms, Measurement, Experimentation

Keywords

Online Community, forum search, thread structure

1. INTRODUCTION

Online communities (e.g., newsgroups, BBS, and forums) are good information sources since knowledge shared by communities has accumulated for years. Search engines, however, have generally overlooked these online community resources. Online community page search results returned from major search engines are often low quality. Internal search services that are provided by the forums are sometimes even worse. One reason for this is that online forum pages are not the same as general web pages. Our goal is to design effective retrieval models that incorporate online forum properties so that the effectiveness of forum search can be improved.

*A full version of this paper is available in [11].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'09 November 2-6, 2009, Hong Kong, China

Copyright 2009 ACM 978-1-60558-512-3/09/11 ...\$10.00.

Online community pages have many unique textual or structural features that distinguish them from general web pages. Generally, a forum has several sub-forums covering high-level topic categories. Each sub-forum has many threads. A thread is a more focused topic-centric discussion unit and is composed of posts created by community members. This is an advantageous feature that encourages in-depth discussion, compared to general web pages.

In this paper, we first annotate thread structures which are based on interactions between community members, using a learning to rank technique. Then, we exploit the thread structure to improve retrieval performance for online community search. Finally, we investigate the effects of threads on retrieval performance.

This paper is unique in combining two strands of work, namely thread structure discovery and retrieval models using the structure. Each subtask has been explored separately in different research areas. The discovery of thread structures in online communities has been addressed by several researchers [6, 13, 2], but their work are limited to specific domains or tasks, e.g., finding question-answer pairs. Retrieval models using multiple contexts or structures are frequently discussed in information retrieval literature [8, 9]. In the literature of social media search, various retrieval models are introduced for each different domain, e.g., blog [4, 10], newsgroup [14], email [12] and forum [5] search. Our work extends these results by exploiting more fine-grained thread structure.

2. DISCOVERY OF THREAD STRUCTURE

A thread is started on some subject by an initiator. If they are interested in the subject of the thread, people post their opinions in reply posts. Because reply posts can reply to any preceding post, many branches (sub-threads) of discussion appear in a thread, and a thread ends up with a tree-shaped structure. We refer to this as a thread structure.

Not all online communities, however, handle threads in the same manner. There are generally two ways that online communities maintain or display threads: flat-view and threaded-view. Flat-view systems, as their name implies, flatten structures of threads and show users all posts in a thread in chronological order. On the other hand, threaded-view systems allow a user to choose a preceding post to reply to, and display posts in structured views.

Currently, flat-view online community pages are still much more prevalent although some online communities emerge that use the threaded view, such as Slashdot¹. One reason for this is that many online forums use popular publishing

¹ <http://slashdot.org/>

software such as phpBB² and vBulletin³ that do not provide a threaded view as a default. Considering the small number of online communities which support threaded views, we believe that techniques for converting flat-view threads to threaded-view threads are needed for online community search. We refer to this conversion as discovery of thread structures.

For simplicity and clarity, we assume that a thread structure is shaped like a rooted tree in which the first post is a root, each child post has only one parent post, and no node is isolated. Under this assumption, when we are given a child post, we can find a reply relation by picking a most likely parent post from among all preceding posts. Constructing a thread structure with reply relations is trivial; thus, finally, our problem is reduced to finding reply relations.

2.1 Intrinsic Features

A straightforward method that we can use to determine a reply relation between two posts is to directly look at the contents of the posts. If two posts address a similar topic, then the posts are likely to have a reply relation. As a similarity measure, we use a variation of the *idf*-weighted cosine similarity, the *Inc.Ltc* term formula [1]. Now we consider which part of a post the similarity measure is applied to.

Quotation vs. Original Content : Many online community systems support an option to quote text from the preceding post when a post is uploaded. Such systems provide split views of the quotation and the original content. Once we obtain the quotation and the original content separately, we can consider various combinations for similarity measurements: i) similarity between the original contents of posts, ii) similarity between the original content of a parent candidate and a quotation of a child post, and iii) similarity between the full text of posts without separating the quotation from the original content.

2.2 Extrinsic Features

A post is an utterance in an informal dialogue. In many cases, a post tends to be short and “instant”. Therefore, similarity features are not often enough to capture relations between posts. Thus, we need to use features which can describe context as well as content. Here we introduce several of these extrinsic features. See [11] for more detailed descriptions about the features.

Location Prior : We assume that a relation between posts can be inferred from the locations of the posts in the chronological time frame. Formally, we want to estimate $P(i_1|i_2)$, that is, the likelihood that a post with location index i_1 is a parent post of a child post with location index i_2 . We estimate this likelihood as a Gaussian Mixture by Expectation-Maximization [11]. We use this likelihood as a feature.

Time Gap : A difference between posting times of two posts can be evidence of a relation between the posts. As a feature, we use a normalized time gap given by $(t_2 - t_1)/(t_2 - t_0)$, where t_0 , t_1 , and t_2 are the posting times of the first post, a parent candidate post, and a child post.

Same Author : Assuming that turn-taking between speakers happens in a thread, the fact that two posts are written by the same author usually can be used as negative evidence

of a relation. We use an indicator of the same author relationship as a feature.

Author Reference : Users often refer to the author of the specific post to express an intention to reply to a specific post. Existence of an author reference between two posts can be used as a feature.

Inferred Turn-taking : Let post A , B and C be posted in this order in a thread. If post A and B have an author reference and post A and C have a same author relation, then we can infer that post C replies to post B when assuming turn-taking with $A \rightarrow B \rightarrow C$. We use the relation between post B and C as a feature.

2.3 Learning

We consider the thread structure discovery task as a ranking task. If each child post is considered as a query, parent candidate posts are considered as documents to be retrieved. Since a post has only one parent post, we have only one relevant document for each query. We use the ranking SVM algorithm [7] to learn the relations.

2.4 Collections

We use three online community collections in order to evaluate techniques for thread structure discovery. See [11] for more detailed descriptions about the collections.

World of Warcraft (WOW) forum : We crawled a discussion forum⁴ of the World of Warcraft (WOW), an online game. This collection contains 16,274 threads. We manually annotated structures of randomly chosen 60 threads.

Cancun forum : We crawled the Cancun forum⁵ of tripadvisor.com, a travel guide site. This collection contains 58,150 threads. The Cancun forum does not systemically support quotation in contrast to the WOW forum. We also manually annotated structures of randomly chosen 60 threads.

W3C email archive : We used the ‘lists’ sub-collection of the W3C collection from the TREC enterprise track [12]. To build an annotation set for thread structure discovery, 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, in this set, we obtained 1,635 threads which contain at least 3 emails.

2.5 Experiments

We conducted experiments for thread structure discovery on each collection. For the WOW and Cancun collections, because the annotated data is small, we performed 10-fold cross validation. Since the W3C collection has enough data for training, i.e. 1,635 threads, we used 1,535 threads as training data and 100 threads as test data. We performed the paired randomization test with $p < 0.05$ to test the statistical significance of improvements.

Table 1 shows the experimental results for the three collections. The techniques which use all features mentioned in the previous sections significantly outperform the baselines which use a single intrinsic feature based on full text. Further, in all collections except the Cancun forum which doesn’t have features based on quotations, the achieved accuracy is very high, i.e. about 90%. Therefore, our approach shows good performance for thread structure discovery when informative features which we introduced are available.

²<http://www.phpbb.com/>

³<http://www.vbulletin.com/>

⁴<http://forums.worldofwarcraft.com/board.html>

⁵<http://www.tripadvisor.in/ShowForum-g150807-i8>

	WOW	CANCUN	W3C
Baseline	0.5858	0.4697	0.8988
Full Features	0.8798	0.6279	0.9617

Table 1: Thread structure discovery results. A number is a recall score, i.e. how many reply relations are correctly detected. ‘Baseline’ uses only content similarity based on full text. ‘Full features’ uses all intrinsic and extrinsic features. All improvements on the baseline are statistically significant.

3. CONTEXT-BASED RETRIEVAL

In this section, we introduce approaches to improve retrieval performance using thread structures discovered by the algorithms introduced in Section 2.

3.1 Context Extraction

A thread contains different self-contained sub-structures. We call a sub-structure a context. Figure 1 presents four contexts. The first context is the coarsest-grained context, i.e. the thread itself. The second context is the finest-grained context, i.e. a post. While we can use thread contexts to get a general picture about the topic addressed by a thread, we can use post contexts to get detailed information. The third context is a pair defined by a reply relation. This context is directly extracted from a relation discovered by thread structure discovery algorithms. A pair context contains an interaction between two users such as a question-answer pair. The fourth context contains all posts in a path from the root node to a leaf node. We refer to this context as a dialogue because by looking at the context we can follow a conversation flow, e.g., how the discussion was started, what issue was discussed, and what the conclusion was. Note that while thread and post contexts are given, pair and dialogue contexts must be extracted from a thread structure.

3.2 Multi-context-based Retrieval

We address two retrieval tasks using multiple contexts: thread search and post search. Since posts in casual online forums such as WOW or Cancun are usually too short to provide information on their own, people are likely to want to find relevant threads rather than posts. On the other hand, emails (posts) in a technical email archive like the W3C archive are often long enough to deliver information. In that case, a more suitable task is to find relevant posts.

For these two tasks, we introduce retrieval techniques based on a language modeling approach to retrieval [3]. In our work, the query likelihood $P(Q|D)$ is estimated by the Dirichlet smoothed unigram language model.

Thread Search

The simplest approach to thread search is to consider a thread as a document [4, 10]. We refer to this as global representation (GR) and estimate $\Phi_{GR}(Q, T_i) = P(Q|T_i)$, where Φ is a ranking function and $P(Q|T_i)$ is a query likelihood score of query Q for thread T_i .

A drawback of GR is that relevant local contexts can be dominated by non-relevant contexts. To tackle this drawback, we consider the pseudo-cluster selection (PCS) [10]. Pseudo-cluster selection retrieves the top N local contexts and aggregates local contexts in the ranked list according to which thread the local context comes from. We call the lo-

cal context group a pseudo-cluster. Finally, relevant threads are located according to a geometric mean of scores of the top K local contexts in a pseudo-cluster as follows:

$$\Phi_{PCS}(Q, T_i) = \left(\prod_{j=1}^K P(Q|L_{ij}) \right)^{1/K}$$

where $P(Q|L_{ij})$ is a query likelihood score based on the language model of local context L_{ij} in thread T_i . PCS has proved effective for thread search based on post contexts [5].

PCS reflects how much relevant information exists locally in a thread whereas GR reflects the cohesiveness of the thread. Therefore, we consider a combination of GR and PCS by $\Phi_{Product}(Q, T_i) = \Phi_{PCS}(Q, T_i)^{(1-\pi)} \cdot \Phi_{GR}(Q, T_i)^\pi$, where π is a weight parameter.

Post Search

If we have post contexts only, a post language model is estimated by smoothing with the collection. If we know that post D belongs to a thread T , then we can do two-stage smoothing as done in cluster-based retrieval [8].

Further, if we have another context X_z , i.e. a pair context or a dialogue context, then we can add one more smoothing stage. However, in contrast to thread contexts, a post can belong to multiple pair contexts or dialogue contexts. We compute a geometric mean to combine language models of the contexts as follows:

$$P_z(w|D) = (1-\lambda_1)P_{ML}(w|D) + \lambda_1((1-\lambda_2)P_{ML}(w|X_z) + \lambda_2((1-\lambda_3)P_{ML}(w|T) + \lambda_3P_{ML}(w|C)))$$

$$P(w|D) = \left(\prod_{z=1}^Z P_z(w|D) \right)^{1/Z}$$

where $P_{ML}(\cdot)$ is a maximum likelihood estimator and Z is the number of local contexts which contain D .

3.3 Experiments

Since the W3C collection has been used for the email discussion search task of the TREC enterprise track, there is a relevance judgment set provided by TREC [12]. On the other hand, we had to make our own relevance judgments for the other two collections. For each collection, we manually created queries from randomly chosen 30 popular thread titles. We created relevance judgment pools using retrieval techniques in Section 3.2 and judged them using a ternary scale. In total, we obtained 2,591 judgments and 2,401 judgments for the WOW and the Cancun collection, respectively.

To extract contexts, we discovered structures for all threads in each collection using the SVM classifier trained with all features in Section 2.

To evaluate performance, we performed 10-fold cross validation. The parameters were exhaustively searched to maximize NDCG@10 for thread search and MAP for post search. We performed the paired randomization test with $p < 0.05$ to test the statistical significance of improvements.

Table 2 shows results of thread search on the WOW and the Cancun collection. ‘Thread’ means GR based on a thread context. ‘Post’, ‘Pair’ and ‘Dialogue’ mean PCS based on each context. ‘+ Thread’ means a combination of GR and PCS. The top three rows which do not need thread structures are considered as baselines.

In the WOW collection, techniques based on dialogue contexts show better or at least comparable performance to

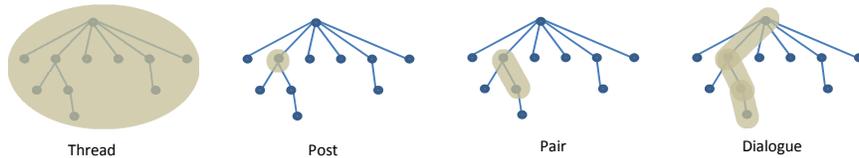


Figure 1: Contexts in a thread structure

	WOW		CANCUN	
	NDCG@10	MAP	NDCG@10	MAP
Thread	0.4200	0.3705	0.4612	0.2630
Post	0.2966	0.2565	0.4763	0.2887
Post+Thread	0.4519	0.3875	0.4942	0.2896
Pair	0.3763 ^β	0.2998 ^β	0.4478	0.2413
Pair+Thread	0.4447 ^{αβ}	0.3885 ^{αβ}	0.4897 ^α	0.2857 ^α
Dialogue	0.4374 ^β	0.3599 ^β	0.4938 ^α	0.2618
Dialogue+Thread	0.4823 ^{αβγ}	0.4073 ^{αβγ}	0.5141 ^{αβ}	0.2973 ^α

Table 2: Retrieval Performance on the WOW and the Cancun collection (Thread Search). The superscripts α , β and γ indicate statistically significant improvements on ‘Thread’, ‘Post’, ‘Post + Thread’, respectively.

	MAP	P@10
Post	0.2405	0.4404
Post+Thread	0.2931	0.4945
Post+Dialogue+Thread	0.3036 ^{αβ}	0.5101 ^{αβ}
Post+Pair+Thread	0.3101 ^{αβ}	0.5147 ^{αβ}

Table 3: Retrieval performance on the W3C collection (Post Search). The superscripts α and β indicate statistically significant improvements on ‘Post’ and ‘Post + Thread’, respectively.

techniques based on the other contexts. Particularly, when using dialogue and thread contexts together, the best performance is achieved, and the improvements over all baselines are statistically significant. This demonstrates that a performance improvement in thread search can be achieved using thread structures, particularly, dialogue contexts.

In the Cancun collection, similar trends are shown, that is, dialogue context-based search and the combination of GR and PCS consistently present better performance than the others. However, the improvements are not always statistically significant. This is presumed to be due to the relative inaccuracy of thread structure discovery in the Cancun collection. To justify this assumption, we investigated retrieval performance based on inaccurate thread structures in the WOW collection. To simulate inaccurate discovery, we used ‘Baseline’ in Table 1 and applied the best retrieval technique, i.e. ‘Dialogue+Thread’. Then, NDCG@10 and MAP are 0.4651 and 0.3869, respectively. This performance is not only worse than the performance based on accurate structure discovery but also fails to show significant differences over the baseline ‘Post+Thread’. This shows that the accuracy of thread structure discovery can be critical in our retrieval framework.

Table 3 shows the results of post search on the W3C collection. Each row represents which contexts are used for smoothing. The one-stage and two-stage smoothing at the top two rows, which use post contexts and threads contexts

only, do not require thread structures. Therefore, we consider them as baselines. For both the pair context and the dialogue context, addition of the thread context for smoothing achieved statistically significant improvements. This shows that contexts based on thread structure are also helpful for post search.

4. CONCLUSION

In this paper, we investigated whether search for community sites such as forums could be improved using thread structure. Our results show that threads can often be accurately identified using our approach. Further, we showed that combinations of multiple thread contexts can achieve significant retrieval effectiveness improvements over strong baselines.

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.

5. REFERENCES

- [1] C. Buckley, J. Allan, and G. Salton. Automatic routing and ad-hoc retrieval using SMART. In *TREC-2*, 1994.
- [2] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *SIGIR '08*, pages 467–474, 2008.
- [3] W. B. Croft and J. Lafferty. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, 2003.
- [4] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and feedback models for blog feed search. In *SIGIR '08*, pages 347–354, 2008.
- [5] J. L. Elsas and J. G. Carbonell. It pays to be picky: an evaluation of thread retrieval in online forums. In *SIGIR '09*, pages 714–715, 2009.
- [6] M. Elsnar and E. Charniak. You talking to me? a corpus and algorithm for conversation disentanglement. In *ACL-08: HLT*, pages 834–842, 2008.
- [7] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, pages 133–142, 2002.
- [8] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *SIGIR '04*, pages 186–193, 2004.
- [9] P. Ogilvie and J. Callan. Hierarchical language models for retrieval of XML components. In *INEX 2004*, 2004.
- [10] J. Seo and W. B. Croft. Blog site search using resource selection. In *CIKM '08*, pages 1053–1062, 2008.
- [11] J. Seo, W. B. Croft, and D. A. Smith. Online community search using thread structure. CIIR Technical Report, UMass Amherst, 2009.
- [12] I. Soboroff, A. P. de Vries, and N. Craswell. Overview of the TREC 2006 enterprise track. In *TREC 2006*, 2007.
- [13] Y.-C. Wang, M. Joshi, W. W. Cohen, and C. Rose. Recovering implicit thread structure in newsgroup style conversations. In *ICWSM II*, 2008.
- [14] W. Xi, J. Lind, and E. Brill. Learning effective ranking functions for newsgroup search. In *SIGIR '04*, pages 394–401, New York, NY, USA, 2004. ACM.