

Adopting Inference Networks for Online Thread Retrieval

Sumit Bhatia* and Prasenjit Mitra*†

*Computer Science and Engineering

†Information Sciences and Technology

The Pennsylvania State University, University Park, PA-16802, USA

sumit@cse.psu.edu, pmitra@ist.psu.edu

Abstract

Online forums contain valuable human-generated information. End-users looking for information would like to find only those threads in forums where relevant information is present. Due to the distinctive characteristics of forum pages from generic web pages, special techniques are required to organize and search for information in these forums. Threads and pages in forums are different from other webpages in their hyperlinking patterns. Forum posts also have associated social and non-textual metadata. In this paper, we propose a model for online thread retrieval based on inference networks that utilizes the structural properties of forum threads. We also investigate the effects of incorporating various relevance indicators in our model. We empirically show the effectiveness of our proposed model using real-world data.

1 Introduction

Online forums offer an interactive platform for information seeking and have become quite popular. They have enabled users to discuss and exchange ideas with one another without any restriction of geographical boundaries. There are thousands of online forums devoted to a broad range of topics. Often these forums have an active community of users helping each other and taking part in discussions. The content is thus being continuously generated “*by the Users, for the Users*”. As an example, consider the online forum of Ubuntu – a popular Linux distribution, that boasts a community of more than 1 million users and more than 1.25 million threads¹. Suppose a user is facing trouble installing sound drivers on his desktop. He can post a question in the forum describing his problem and other users then post solutions to the problem. Previous work has focused on this aspect of online forums and leveraged it for question answering purposes (Bian et al. 2008; Cong et al. 2008; Hong and Davison 2009).

However, online forums are not just limited to factual questions and answers. Oftentimes, the questions which users ask in a forum do not have a precise answer. As an example, consider a person who is planning to buy an MP3 player but is not sure which brand to purchase. He posts his question in a forum related to electronic devices and gets

opinions of other users. Contrast this case with the first example. There was a fixed procedure for installing sound drivers, but there is no fixed answer to the brand of MP3 players one can purchase. The user here might have certain requirements and might be looking for certain features. He seeks other users’ views, opinions and experiences to determine a suitable MP3 player for himself.

The above example illustrates another distinctive feature of online forums – *discussions*. In forums, people discuss and share their opinions about various topics, news, events as well as about pros and cons of various products, movies, sports etc. Past discussions result in a tremendous volume of data getting stored in forum archives. In that case, a vertical search engine can help the user search for past discussions/solutions without having to post his problem and wait for replies from other users. Such kinds of information needs are hard to satisfy by question answering systems and motivate the need for developing techniques to efficiently manage and search the huge amount of information that the online forums provide.

Searching online forums however, requires special techniques. Current search techniques are not optimized for web forums. One major assumption for generic web search is that each individual web page is a *self-contained* document. However, this assumption does not hold for forum search. When a user starts a topic by posing a question or initiating a discussion, the replies of other users are arranged in a thread-like structure. If the discussion is long, a thread may stretch over more than one webpage and possibly over tens of webpages. As an example, consider a thread which runs over two pages. The thread starter has asked a question and different users are posting answers to the question. Suppose the correct answer to the question is present on the second webpage. If we present only this page to the user then the whole context of discussion, which might be extremely important, is lost. Moreover, using only a single page (either first or second) to determine relevance to a user query fails to utilize the information present in other pages. Thus for forum search, the appropriate unit to use to compute relevance is the whole discussion thread.

Another important difference between generic web pages and forum pages is that forums have a unique structure. A forum is composed of multiple threads and threads have multiple posts. Threads have titles and in some forums, indi-

vidual posts can have titles. Posts also have authors associated with them. Utilizing this structure and associated meta-data can provide a better search functionality than that enabled by using only the textual content of the forum threads.

Wilkinson (1994) has shown that utilizing structure of documents helps improve retrieval performance. However, as discussed in next section, most of the present approaches for forum search have ignored this aspect. In addition, forum pages also have certain unique characteristics that help in indicating the *usefulness/relevance* of a given thread. Further, different forums may have different characteristics and different queries may require different strategies to identify relevant threads. Identifying how much to weigh each individual piece of information (e.g., title, authority of poster, etc.) and how to vary the weights for different types of forums and queries is a challenging problem. In this work, we focus on the former sub-problem, the latter problem being the focus of future work. The formalism of Bayesian Networks affords a powerful and theoretically sound framework to utilize and combine multiple evidences for document ranking. We employ Inference Networks to efficiently combine evidences from different structural units of forum threads as well as for incorporating various prior evidences while computing document relevance probabilities. Our model utilizes multiple attributes like the initial post and the replies, the length of the posts, an authority measure of the posters and the links between posts in a unified inference network based framework to rank threads in response to user queries.

2 Related Work

As identified by Xu and Ma (2006), as much as 75% of the links on a typical forum page points to noise pages like user profiles, login pages etc. and are not actually *recommendations*. Hence, link based algorithms like PageRank and HITS can not be used effectively. They build topic hierarchies and combine the links induced by content similarity with the document link graph for ranking of forum pages. However, it is difficult to categorize the large number of online forums with a single topic hierarchy. Chen *et al.* (2008) propose a ranking algorithm called Posting Rank to rank forum posts. Their approach is based on the assumption that users participating in common threads tend to share common interests and such threads are topically related to each other. However, this assumption might not be true in all the cases. It is fairly common for a single user to have varied interests and participate in threads on different topics. In both the above works, the unit of retrieval is individual forum pages. However, discussions in web forums are organized into threads which usually span many web pages. Hence, retrieval at individual page level ignores the context of discussions going on in the thread and might not be able to capture the relevance to user's information needs completely. Elsas *et al.* (2009) are among the first to review strategies for thread retrieval on a test collection of 48 <query, relevant document> pairs. Seo *et al.*, (2009) describe how the reply structures in forum threads can be recovered and utilized for thread and post level retrieval. Our work on the other hand utilizes inference networks for evidence combination and in addition, also explores the effect of incorporat-

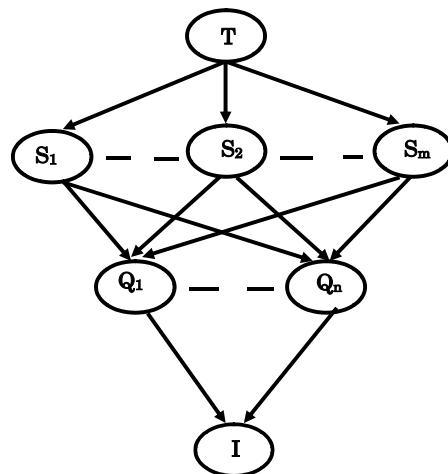


Figure 1: Inference Network Model for Forums

ing various non textual relevance indicators that may help in an improved ranking of returned results.

3 Problem Formulation

We define a *post* as the smallest unit of communication in online forums that consists of content posted by a user. Each post has associated meta-data such as user ID of the user posting the post, time of posting the post etc. *InitPost* I is the initiating post of a thread which is the first post in the thread and initiates discussions. All other posts in a thread are the *ReplyPosts* posted by users participating in the discussion started by the *initPost* I . A thread can now be defined as a triple $T = \langle t, I, \mathcal{R} \rangle$, where t is the title of the thread, I is the *InitPost* and \mathcal{R} is the set of reply posts. Note that \mathcal{R} can be empty in case no user has replied to the *InitPost*. Given a user query Q , our task is to return a ranked list of threads $L = T_1, T_2, \dots, T_n$ to the user such that for all $1 \leq i, j \leq n$ and $i < j$, $rel(T_i, Q) \geq rel(T_j, Q)$, where $rel(T, Q)$ is an appropriate measure of relevance of thread T to query Q . We assume that each thread T deals with a single topic of discussion.

4 Inference Network for Web Forums

As described by Turtle and Croft (1991), the information retrieval problem can be considered as an inference problem where given the event that a document D is observed, one *infers* the probability that the user's information need is satisfied by the document. Metzler and Croft (2004) have shown how language models and inference networks can be combined together for information retrieval purposes. Our implementation also uses the INDRI² toolkit developed by them.

The proposed retrieval model for web forum search is described in Figure 1. The leaf node I is the information need node and represents the event that the user has the need for

²<http://www.lemurproject.org/indri/>

information I . The root node T is the thread node and represents the event that the thread T has been observed. Note that in the actual network there is one such thread node for each thread in the collection. We show only one thread node for the sake of clarity. We want to estimate the probability $P(T|I)$ that the thread T satisfies the information need I . Using Bayes rule, we get:

$$P(T|I) \stackrel{rank}{=} P(T) \cdot P(I|T) \quad (1)$$

Here $\stackrel{rank}{=}$ indicates that the L.H.S and R.H.S have the same rank ordering. $P(T)$ is the prior probability that thread T is relevant for the information need I . The second term represents the probability $P(I|T)$ that the user’s information need I is satisfied by the thread T . The information need I is represented by the user in terms of the query Q consisting of n query terms Q_1, Q_2, \dots, Q_n . This is represented by *query term nodes* Q_1, Q_2, \dots, Q_n in Figure 1. Using the independence assumption for query terms, we get:

$$P(T|I) \stackrel{rank}{=} P(T) \prod_{i=1}^n P(Q_i|T) \quad (2)$$

Here $P(Q_i|T)$ represents the likelihood that the query term Q_i is generated by the thread T . We use the thread structure in order to utilize these query term likelihoods. After the thread node, we use separate nodes, S_1, S_2, \dots, S_m to represent each of the m structural components of the thread. As described above, a typical forum thread consists of three structural elements namely title, InitPost and \mathcal{R} , the set of reply posts and we have a S node for each of these three structural units. Also note that there might be some reply posts in longer threads that causes the topic of discussion to change (*topic drift*). There might also be some *noisy* posts that provide redundant or no new information. The S node corresponding to reply posts can be replaced by three separate nodes representing *off topic* posts, *noisy* posts and posts related to the original topic of discussion. Such a post level analysis and thread segmentation is out of the scope of present work and it is assumed that each thread deals with a single topic of discussion.

We estimate $P(Q_i|T)$ in terms of $P(Q_i|S_{j_T})$, i.e., the likelihood that the query term is generated by the j^{th} structural component S_{j_T} of thread T . Using the total probability rule, above equation yields:

$$P(T|I) \stackrel{rank}{=} P(T) \prod_{i=1}^n \left\{ \sum_{j=1}^m \alpha_j P(Q_i|S_{j_T}) \right\} \quad (3)$$

where α_j determines the weight given to component j and $\sum_{j=1}^m \alpha_j = 1$.

The above equation is used as the ranking function in our model. Given an information need I represented by a query Q , we compute $P(T|I)$ for each thread in the collection and rank them in decreasing order of $P(T|I)$ values. The α parameters control the weightage given to each structural component and can be determined experimentally as explained in Section 6.

In order to estimate the likelihoods $P(Q_i|S_{j_T})$ we use the standard language modeling approach in information retrieval (Ponte and Croft 1998) with *Dirichlet Smoothing* as follows:

$$P(Q_i|S_{j_T}) = \frac{f_{Q_i, j_T} + \mu \frac{f_{Q_i, j_C}}{|j|}}{|j_T| + \mu} \quad (4)$$

Here,

f_{Q_i, j_T} = frequency of term Q_i in j^{th} structural component of thread T ,

f_{Q_i, j_C} = frequency of term Q_i in j^{th} structural component of all the threads in the collection C .

$|j_T|$ is the length of j^{th} structural component of thread T ,
 $|j|$ is the total length of j^{th} structural component of all the threads in the collection C ,

μ is the Dirichlet smoothing parameter. In this paper we set μ to be equal to 2000, a value that has been found to perform well empirically (Zhai and Lafferty 2001).

5 Utilizing Prior Evidences

Online forums contain various important indicators of *quality* or *usefulness* of a particular thread which are not captured by the likelihood measures. Most of such indicators are query independent and hence, provide an *a priori* estimate of a thread’s usefulness. In this section, we describe three such measures.

5.1 Length of Thread

In general, document length is a useful measure of document content. Likewise, for a given query we expect longer threads to provide more information to the user than a smaller thread. We define the length of a thread to be the number of replies in the thread instead of the number of words as is normally the case when measuring document lengths. This is because a larger number of posts indicate more participation as well as more user generated content. Number of replies in a thread is thus a measure of “popularity” as well as “usefulness” of the thread. We also note that this observation may not always be true. Sometimes, a thread with only one reply might provide all the relevant information whereas a very long thread may fail to do so. We do not consider the effect of such exceptional cases for the time being. We model the thread prior based on the number of reply posts as follows:

$$P(T|len) = \lambda_{length} |\mathcal{R}(T)| \quad (5)$$

where, λ_{length} is a normalizing constant not useful for ranking, $\mathcal{R}(T)$ is the set of reply posts in thread T .

5.2 User Authority

In a web forum, both novice users as well as expert or advanced users participate. Generally, novice users ask questions and advanced users provide solutions to their problems. Similarly, in discussion forums/review forums, views and opinions of expert users are more important than that of novice users.

Figure 2 illustrates the behavior of *novice* and *expert* users as observed from one of the datasets used in this paper³.

³Due to space constraints, we show the figure for only one of the

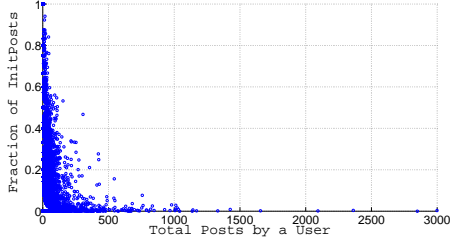


Figure 2: Plot showing user posting behavior

Each user is represented by a point on the scatter plot with total number of posts by a user on X axis and fraction of those posts which were `initPosts` on Y axis. It is clear from the figure that a majority of users has a smaller number of total posts and a higher fraction of `initPosts`, a behavior typical of an *information seeker*. On the other hand, there is also a small clique of users with a large number of total posts and almost no `initPosts`. These are the users that act as *information providers*. Hence, postings made by a user are an indication of his “authority” and as such content provided by such “authoritative” users is more important.

We define the authority $A(u)$ of user u as:

$$A(u) = \lambda_{auth} \left\{ \frac{N_p(u) - N_{ip}(u)}{N_p} + \frac{1}{N_u} \right\} \quad (6)$$

where,

$N_p(u)$ is the total number of posts by user u ,
 $N_{ip}(u)$ is the total number of `initPosts` by user u ,
 N_p is the total number of posts in the collection,
 N_u is the total number of users and
 λ_{auth} is a normalizing constant.

In the above equation, the first term inside the bracket measures the contribution of the user to all the replies in the collection. As already explained, *information provider* contributes more than an *information seeker*. The second term acts as a smoothing parameter and assigns a uniform default authority to each user in the collection. Using $A(u)$, a simple measure of *authority of a thread* $A(T)$ can be defined as:

$$A(T) = \frac{1}{|T|} \sum_{p=1}^{|T|} A(u_{pT}) \quad (7)$$

Here, $|T|$ is the number of posts in thread T and $A(u_{pT})$ is the authority of the user posting the post p in T . This definition measures authority per user in thread T . Therefore, discussions in which more *authoritative* or *expert* users participate are given a higher weight because we expect content generated by such users to be of high quality. Once we have $A(T)$, we can use it to model the prior probability as:

$$P(T|A(T)) = \lambda_{authority} A(T) \quad (8)$$

Again, $\lambda_{authority}$ is a normalizing constant, not useful for ranking.

datasets(Ubuntu). Similar behavior was observed for the second dataset used in this work.

5.3 Link Information

Oftentimes, users provide links to related threads in an ongoing discussion. Such links to other threads provide certain evidence that the linked-to thread contains some useful relevant information with respect to the ongoing discussions or contains a solution to the problem asked in the current thread. In order to utilize these link-based evidences, we define the *Forum Graph* $F(\mathcal{T}, \mathcal{E})$ where node set \mathcal{T} consists of all the threads in the collection, each thread being a node in the graph and \mathcal{E} is the edge set consisting of directed edges. An edge (T', T) exists iff there is a post p in thread T' that links to thread T . The weight of the edge (T', T) is $A(u_p)$, i.e., the authority of user posting the post p . Thus, recommendations provided by an *expert* user are considered more important. This phenomenon is illustrated in Figure 3. By constructing the forum graph in this way, we can compute a simple *Link Score* for thread T (which is then used as an estimate of its prior probability) as follows:

$$L(T) = \sum_{T': T' \rightarrow T} weight(T', T) \quad (9)$$

which yields

$$P(T|L(T)) = \lambda_{link} L(T), \quad (10)$$

λ_{link} being the normalizing constant.

Note that the above definition of forum graph allows multiple edges between T' and T in case more than one user in T' links to T . However in the datasets used in this paper, we did not find any such case.

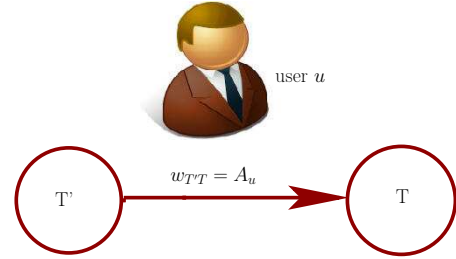


Figure 3: Presence of user provided evidence based links in forum threads.

6 Experiments and Results

6.1 Data Description

In order to evaluate the proposed models we used threads crawled from the following two online forums:

1. Ubuntu Forums: Official forum of the Ubuntu Linux distribution⁴
2. TripAdvisor – NewYork: Popular forum for travel related discussions⁵.

For each forum, all the crawled web-pages were pre-processed and all the pages belonging to the same thread were identified and processed to identify different structural

⁴<http://ubuntuforums.org>

⁵http://www.tripadvisor.com/ShowForum-g28953-i4-New_York.html

units and their associated metadata (title, posts, user IDs etc.). Stemming was performed using Porter’s stemmer and stop words were removed using a general stop word list of 429 words used in the Onix Test Retrieval Toolkit⁶. Table 1 summarizes various descriptive statistics of the two datasets.

	Ubuntu	TripAdvisor
Total No. of Threads	113277	83072
Total No. of Users	103280	39454
Total No. of Posts	676777	590021
Average Thread Length (in no. of posts)	5.98	7.10
Total No. of interlinks	2899	4589

Table 1: Summary statistics of different datasets used. Interlinks refer to the number of interthread links with in a dataset.

6.2 Experimental Protocol

Since the relevance judgments were not available for the two datasets used, we took help of two human annotators to create relevance judgment pools for the two collections. We assigned ternary relevance judgments to each thread for a given query – 0 for totally irrelevant threads, 1 for partially relevant and 2 for highly relevant threads. The search page of ubuntu forums provides a list of 70 most searched for terms which were used to generate queries for our experiments. Similarly, tripadvisor forum provides a list of most frequently searched for topics. The queries for this dataset were generated by extracting keywords from the frequently searched for topics. In total, we generated 25 queries for each dataset. Some example queries for each dataset are listed in Table 2. For each query, all the threads returned by various methods were then assigned relevance judgements scores. In all, relevance judgments were assigned for 4512 threads in the Ubuntu dataset and 4478 threads in the TripAdvisor dataset. In order to compare the performance of various retrieval methods, we report Mean Reciprocal Rank (MRR), Precision @ 10, NDCG @ 10 and Mean Average Precision (MAP).

6.3 How Structure Helps?

Our first set of experiments aims at finding the relative importance of each individual structural component in the retrieval process. We also investigate if utilizing and combining evidence from different structural components help in improving retrieval performance.

In order to investigate these issues, we first chose a model constructed from the whole thread as our baseline (wholeThread). This model does not differentiate between different structural components of a thread and is obtained by building a standard language model for each thread using equation (4). Next, we tested retrieval models that utilize information only from one of the structural components of a forum thread. As described above in section 3, we can

Dataset	Example Queries
Ubuntu	mount ntfs partition ubuntu firefox flash no sound
TripAdvisor	how to get to statue of liberty safety in manhattan

Table 2: Some example queries used in the experiments.

Method	MRR	P@10	NDCG@10	MAP
Ubuntu Dataset				
wholeThread	0.6357	0.2920	0.6143	0.5115
T	0.2000	0.0920	0.2445	0.1683
I	0.3518	0.1240	0.4088	0.3190
R	0.5091	0.2480	0.5624	0.4467
T+I+R (.75,.1,.15)	0.6580	0.3360	0.6308	0.5397
TripAdvisor Dataset				
wholeThread	0.6445	0.4880	0.6457	0.5865
T	0.2524	0.0720	0.2885	0.2317
I	0.2687	0.0840	0.3369	0.2630
R	0.6478	0.3920	0.6655	0.5664
T+I+R (.6,.2,.2)	0.7351	0.5000	0.6882	0.6326

Table 3: Impact of different structural components and their combinations on retrieval performance on the two datasets. For the T+I+R model, numbers in brackets denote the optimal α values for the respective components.

assign differential weights to evidence from different components by choosing suitable values for the corresponding α parameter of each component. Therefore in order to have a model based on only one structural unit, we set the α parameter for that component equal to 1. α parameters for all other components are set to 0. This gives us three baseline retrieval models, one each for Title(T), initPost(I) and replyPosts(R). Next, in order to have models that combine evidence from different components, we need to find suitable α values for each component. We varied the weights given to each structural component in intervals of 0.05 with the constraint of equation (3) that the sum of weights is equal to one ($\alpha_1 + \alpha_2 + \alpha_3 = 1$). Since the end-users of a search engine want the relevant results in top few documents, the parameters were selected using five folds cross validations to optimize the *Precision @ 10* metric. The experimental results are summarized in Table 3. The results reported are averaged over 5 folds.

Comparing the three single structure models (T , I and R), we find that the model based on the Reply Posts outperforms the other two models across all evaluation metrics for both the datasets. This behavior is expected because the replies constitute the major portion of a thread. An interesting observation that can be made is that the performance of title model is worst among the three structural components. Many of the online forums, including the one used in this work (Ubuntu Dataset), offer facilities to search for threads using title only. The basic intuition behind such an approach is that one expects the thread starters to give meaningful titles to their threads in order to get suitable replies from other users. But as our results show, this strategy is suboptimal. This is because many times the title fails to capture the com-

⁶<http://www.lextek.com/manuals/onix/stopwords1.html>

Method	MRR	P@10	NDCG@10	MAP
Ubuntu Dataset				
T+I+R	0.6580	0.3360	0.6308	0.5397
Length (L)	0.7333	0.4560	0.7363	0.6247
Authority (A)	0.6941	0.4560	0.6935	0.5931
Link (Li)	0.7567	0.4680	0.7430	0.6622
L+A+Li	0.6020	0.4200	0.6688	0.5177
TripAdvisor Dataset				
T+I+R	0.7351	0.5000	0.6882	0.6326
Length (L)	0.7140	0.5240	0.7162	0.6487
Authority (A)	0.7144	0.5520	0.7277	0.6730
Link (Li)	0.8067	0.5840	0.7382	0.6877
L+A+Li	0.6957	0.5240	0.6864	0.6205

Table 4: Effect of incorporating various priors information indicators.

plete information about the thread due to its short length. Also, often the topic of discussion deviates from the original topic as the thread progresses. We also observed many titles in our dataset that did not convey any useful information about the thread (eg. “save me, computer not working”). These factors collectively result in a comparatively poor performance of the title as well as the initial post model. The real strength of our approach is illustrated by the retrieval model with optimized weight parameters that conclusively outperforms all other models across all the evaluation metrics. We achieve higher precision as well as higher MRR and NDCG values that means that not only we are getting more relevant results, we are getting them at lower ranks (or higher up in the returned results) which is expected of any practical information retrieval system. These results confirm our hypothesis that utilizing structure of forum threads helps improve the retrieval performance and maximum gains are achieved when we utilize information from all structural components.

6.4 Effect of Incorporating Priors

In this section we study the effect of incorporating various prior evidences that indicate the relevance of a thread. We have described three such measures - length of the thread, user authority and link information. These measures can be incorporated as prior probability $P(T)$ in equation 3. We use the optimized model from previous section as a baseline and study the effect of adding priors to this model. The results are summarized in Table 4.

One general observation is that by incorporating each of the three priors individually, the resulting models outperform the fine-tuned baseline model across all the evaluation metrics. Incorporation of length prior results in the inclusion of documents containing more discussions about a topic and thus has a higher chance of being relevant. The authority prior assigns higher weights to documents that contain content posted by experienced users which is generally, of high quality. Further, the performance of the link prior is best among all the models across all the evaluation metrics. This is because the link prior captures the intra-forum relationships and utilizes the evidences provided by the forum users about the usefulness and relevance of forum threads.

Further, evidences provided by expert users are considered more important than those provided by novice users.

One counter-intuitive result that we obtain is that when we combine all the three priors, the performance deteriorates. The combined model underperforms the models that utilize only a single prior. This indicates that for some of the queries, gains achieved by using a single prior are lost when we add additional prior information. Thus identifying what priors are useful for what types of queries is necessary and will be a part of future study.

7 Conclusions and Future Work

In this paper we discussed how the problem of search in online forums is different from generic web search and proposed an inference network based retrieval model to utilize the rich structure of forum threads. Experiments with a baseline retrieval model showed that utilizing structural information of forum threads helped improve retrieval performance. It was shown that incorporating various non-textual sources of information helps in boosting the ranking of search results. Our future work will focus on identifying topic drifts in longer threads and summarization of discussions in forum threads.

8 Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Nos. 0535656 and 0845487. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We would like to thank Sunil Viraj Jain for his help in conducting the experiments.

References

- Bian, J.; Liu, Y.; Agichtein, E.; and Zha, H. 2008. Finding the right facts in the crowd: factoid question answering over social media. In *WWW '08*, 467–476.
- Chen, Z.; Zhang, L.; and Wang, W. 2008. Postingrank: Bringing order to web forum postings. In *AIRS 2008*, volume 4993, 377–384.
- Cong, G.; Wang, L.; Lin, C.-Y.; Song, Y.-I.; and Sun, Y. 2008. Finding question-answer pairs from online forums. In *SIGIR 2008*, 467–474.
- Elsas, J. L., and Carbonell, J. G. 2009. It pays to be picky: an evaluation of thread retrieval in online forums. In *SIGIR '09*, 714–715.
- Hong, L., and Davison, B. D. 2009. A classification-based approach to question answering in discussion boards. In *SIGIR '09*, 171–178.
- Metzler, D., and Croft, W. B. 2004. Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.* 40(5):735–750.
- Ponte, J. M., and Croft, W. B. 1998. A language modeling approach to information retrieval. In *SIGIR 1998*, 275–281.
- Seo, J.; Croft, W. B.; and Smith, D. A. 2009. Online community search using thread structure. In *CIKM 2009*, 1907–1910.
- Turtle, H., and Croft, W. B. 1991. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems* 9(3):187–222.
- Wilkinson, R. 1994. Effective retrieval of structured documents. In *SIGIR 1994*, 311–317.
- Xu, G., and Ma, W.-Y. 2006. Building implicit links from content for forum search. In *SIGIR 2006*, 300–307.
- Zhai, C., and Lafferty, J. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR 2001*, 334–342.