Exploiting Thread Structures to Improve Smoothing of Language Models for Forum Post Retrieval

Huizhong Duan and Chengxiang Zhai

University of Illinois at Urbana-Champaign,
201 N Goodwin Ave, Urbana, IL 61801, USA
duan9@illinois.edu,czhai@cs.uiuc.edu

Abstract. Due to many unique characteristics of forum data, forum post retrieval is different from traditional document retrieval and web search, raising interesting research questions about how to optimize the accuracy of forum post retrieval. In this paper, we study how to exploit the naturally available raw thread structures of forums to improve retrieval accuracy in the language modeling framework. Specifically, we propose and study two different schemes for smoothing the language model of a forum post based on the thread containing the post. We explore several different variants of the two schemes to exploit thread structures in different ways. We also create a human annotated test data set for forum post retrieval and evaluate the proposed smoothing methods using this data set. The experiment results show that the proposed methods for leveraging forum threads to improve estimation of document language models are effective, and they outperform the existing smoothing methods for the forum post retrieval task.

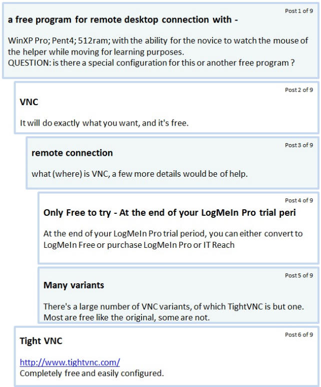
Keywords: forum post retrieval, language modeling, smoothing

1 Introduction

There are nowadays more and more ways for publishing information on the Web. Among them, online forums and discussion boards are of great importance and widely used. The reason lies in several aspects. For one thing, it is much easier for users to post contents on forums, compared with composing web pages. The infrastructure of forums allows users to focus on the content of the post instead of putting much effort on the designing of the presentation. For another, users are able to interact with each other in forums while they publish their opinions. This makes the web contents live and people are therefore more inclined to looking into forum posts for information. As more and more forums are available online, forum post retrieval becomes an important task. According to one of the most popular forum search engines, BoardTracker, it has more than 32,000 forums indexed [1]. A number of forum search engines have been built in recent years[1][2][3]. Despite the growth of forums, little research has been done on models for forum post retrieval.

As a new retrieval problem, forum post retrieval both raises new challenges and offers new opportunities. The challenges are raised from the unique characters of forum posts. Posts are usually short in length. Background information is often omitted in posts as it is assumed that readers share the same background knowledge. For example, in Figure 1, post 2 and post 6 are suggesting the software "VNC" without mentioning its usage. This is because the authors of the posts assume their readers have already read the previous posts and are hence aware of the topic they are talking about. This raises big challenges for traditional retrieval techniques as the relation among posts cannot be overlooked.

On the other hand, there are new opportunities to optimize the performance of forum post retrieval as forums contain richer context for each post. A post usually has strong connection with its previous discussions. Therefore, the thread structure can be leveraged to overcome the aforementioned problems and improve the accuracy of retrieval. Indeed, recent work [13] has shown that high-quality thread structures learned based on manually created training data can improve retrieval performance. However, it is still unclear how we can improve retrieval accuracy by using only the *raw* thread structures naturally available in all the forums. In this paper, we study the use of raw structure information of forum contents to improve the quality of retrieval. Particularly, we study how to use the thread structure to improve smoothing of language models for forum post retrieval.



Post 1 of 9

a free program for remote desktop connection with -

WinXP Pro; Pent4; 512ram; with the ability for the novice to watch the mouse of the helper while moving for learning purposes.
QUESTION: is there a special configuration for this or another free program ?

Post 2 of 9

VNC

It will do exactly what you want, and it's free.

Post 3 of 9

remote connection

what (where) is VNC, a few more details would be of help.

Post 4 of 9

Only Free to try - At the end of your LogMeIn Pro trial peri

At the end of your LogMeIn Pro trial period, you can either convert to LogMeIn Free or purchase LogMeIn Pro or IT Reach

Post 5 of 9

Many variants

There's a large number of VNC variants, of which TightVNC is but one. Most are free like the original, some are not.

Post 6 of 9

Tight VNC

http://www.tightvnc.com/
Completely free and easily configured.

Fig. 1. A Fragment of a Thread

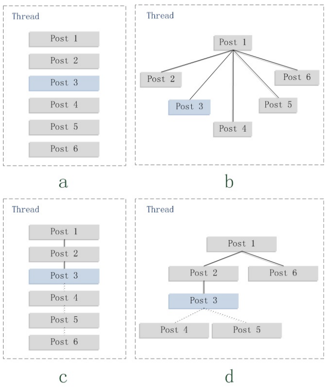


Fig. 2. Representation of a Thread

[1] http://www.boardreader.com

[2] http://www.boardtracker.com

[3] http://www.omgili.com

We propose two smoothing schemes for exploiting the thread structure in forums. The first is model expansion, which is in essence a variant of the two stage smoothing scheme [20]. In the first stage, it makes use of the language models of related posts to smooth the language model of the target post, in order to expand the language model to incorporate more contextual information and give better estimation of the topical words. In the second stage, it uses the collection language model to explain away the common words in queries. The second scheme is count expansion. In this smoothing scheme, we directly propagate the counts of words from relevant posts within the same thread to the target post. Then the language model is estimated based on the propagated word counts using maximum likelihood estimation. The model is further smoothed with a collection language model in the end. Experiments show that both smoothing schemes can achieve good performances through appropriate use of thread structures.

Within the two proposed schemes, we further study different ways of adopting the thread context for a given post to improve the estimation of language models. Particularly, we study four different representations of the thread structure, namely the flat plate representation, one level tree representation, timeline representation and reply tree representation. Moreover, we also study different weighting functions for combining the contents of relevant posts, including structural distance, content similarity as well as their combination. Our experiments show that smoothing with the reply tree representation and combined weighting function tend to consistently achieve the best performance.

To test the performance of our proposed methods as well as existing retrieval models, we created a test set consisting of a full crawl of an online forum as well as a set of automatically generated queries. The queries are generated from the online community of question answering services to reflect the real world information needs. Manual judgements are obtained through the use of Amazon Mechanical Turk[4]. Voting is performed to ensure the quality of judgements. The test set has been made publicly available for future research on this topic[5].

2 Related Work

Our study is based on the state-of-the-art language modeling approach in information retrieval [6][19]. Language model was first applied to information retrieval more than a decade ago [11][9][4]. During its development, smoothing was shown to be a crucial part for achieving good retrieval performance [19][20]. Furthermore, Liu and Croft [8] and Tao et al. [14] used richer context for smoothing to further improve the performance. Our work adds to this line of work a new way of smoothing based on thread structures.

One similar topic to forum post retrieval is XML retrieval. XML retrieval is similar to forum post retrieval but different in the following aspects. First, an XML document is usually composed by the same author instead of multiple authors. Therefore, such contents are usually better formatted and more

[4] https://www.mturk.com

[5] http://timan.cs.uiuc.edu/downloads.html

grammatical. Second, XML retrieval does not require long queries for expressing information need, as XML documents are typically records of facts and events. Queries for such information are usually short in length. For XML retrieval, language modeling was also shown to be effective [3][10]. Ogilvie and Callan introduced shrinkage models which takes each node's parent node's model in estimation of language model [10]. Our method of modeling with reply structure is similar to the shrinkage method, but does not require as many parameters. Therefore, it is more suitable for practical use. Another related body of literature is Email discussion retrieval. Weerkamp et al. used thread context to improve language modeling for Email archive retrieval [15]. However, they did not research into the detail structure of threads. In this paper we study the internal structure of forum threads for improving the performance of forum post retrieval.

There are also some initial work on forum mining and retrieval. Lin et al. [7] used a sparse coding approach to model the semantic topics of forum posts, and applied the model to reconstruct the reply relationship between posts. Xu and Ma [16] built implicit links among posts to simulate the link based algorithm on web pages. Cong et al. [2] and Hong and Davison [5] extract question answer pairs from discussions. None of these work studied the problem of forum post retrieval directly. One of the most recent work [13] revealed that with the *accurately annotated* structure of thread, retrieval performance can be substantially increased; in contrast, we study the potential of using *raw* thread structures in the threads to improve retrieval performance.

3 State-of-the-Art Language Models for Information Retrieval

Language model has been extensively studied for information retrieval in recent years. In this work, we use the KL-Divergence model [18] as our base model. In this model, queries and documents are considered as samples generated from different language models, and the ranking of documents is based on the negative KL-Divergence of the query language model and the document language model. Formally, the ranking score is computed as:

$$-D_{KL}(\theta_q||\theta_d) = -\sum_w p(w|\theta_q) log \frac{p(w|\theta_q)}{p(w|\theta_d)} \tag{1}$$

where θ_q and θ_d are query language model and document language model, respectively. They are usually estimated through maximum likelihood estimation. Note that KL-Divergence is a general form of language model as it naturally subsumes other models such as query likelihood model.

To avoid overfitting and zero probability problem, smoothing is needed. We make use of two commonly used smoothing methods: Jelinek Mercer (JM) smoothing and Dirichlet prior smoothing [19]. In JM smoothing, each document language model is linearly interpolated with a collection background model:

$$p(w|\hat{\theta}_d) = (1 - \lambda)p_{ml}(w|\theta_d) + \lambda p_{ml}(w|C) \tag{2}$$

where $p_{ml}(w|\theta_d)$ is the maximum likelihood estimation of the probability of word w in document d's language model, $p_{ml}(w|C)$ is the probability of w in the background model. λ is a parameter controlling the amount of smoothing.

In Dirichlet smoothing, the document model is considered to have a conjugate prior with Dirichlet distribution. The derived smoothing formula is:

$$p(w|\hat{\theta}_d) = \frac{|d|}{|d|+\mu} p_{ml}(w|\theta_d) + \frac{\mu}{|d|+\mu} p_{ml}(w|C) \tag{3}$$

where μ is a parameter controlling the amount of smoothing and can also be interpreted as the total number of pseudo counts of words introduced through the prior.

A number of techniques were proposed to further improve these basic smoothing methods [17]. Closely related to our work are two studies [8][14] that use similar documents to smooth a document language model. Liu and Croft[8] proposed a clustering based smoothing method (CBDM), where each document is first smoothed with clustering analysis before being smoothed with the collection language model. Tao et al.[14] improved CBDM by introducing the neighborhood based smoothing method (DELM). In their method, each document is enriched using the contents of the documents that have the highest content similarity with it before apply maximum likelihood estimation. Therefore documents are treated more fairly in terms of incorporating contexts. Both methods are expensive to compute. Although these methods can be applied to forum post retrieval, they do not take advantage of the thread structures in a forum. As shown in Figure 1, a thread provides very useful context for a post. In the next section, we propose new smoothing methods to exploit thread structures for smoothing.

4 Improving Smoothing By Exploiting Thread Structure

In this section, we propose two smoothing schemes for exploiting thread context to improve smoothing of a forum post language model, namely the model expansion scheme and the count expansion scheme. We then study different ways of exploiting thread context, as well as ways to weight and combine them into our smoothing scheme.

4.1 The Smoothing Schemes

Model Expansion. The essence of our model expansion scheme is a two stage smoothing process. In the first stage we make use of language model of the thread context of a post to smooth its language model obtained through maximum likelihood estimation. This is meant to improve the coverage of contextual words as well as to improve the estimation of topical words. In the second stage we further smooth the language model with a reference model, in order to assign non-zero probability to unseen words and explain away the common words [20]. In the first stage of smoothing, Dirichlet prior smoothing is used because the amount of contextual information to be incorporated is dependent on the length of the target post. A short post would need more contextual contents while a long post may already have enough contexts in itself. In the second stage Jelinek Mercer smoothing is used. In fact, different settings of smoothing methods are explored in practice and it is concluded that such a setting consistently achieves a slightly better performance. Therefore, we base our discussion on this setting. With the KL-Divergence language model, the ranking formula of model expansion scheme can be derived as:

$$-D_{KL}(\theta_q||\theta_d) \propto \textstyle\sum_{w\in d} p_{ml}(w|\theta_q) log(1 + \frac{(1-\lambda)(\frac{|d|}{|d|+\mu} p_{ml}(w|\theta_d) + \frac{\mu}{|d|+\mu} p(w|\theta_{T(d)}))}{\lambda p_{ml}(w|C)}) \tag{4}$$

where

$$p(w|\theta_{T(d)}) = \sum_{d'\in T(d)} \omega(d',d) p_{ml}(w|\theta_{d'}) \tag{5}$$

where $T(d)$ is the thread context/relevant content of d, and $\omega(d',d)$ is a weighting function for the interpolation of the thread contexts, $\sum_{d'\in T(d)} \omega(d',d) = 1$. Given this smoothing scheme, our major challenge is to figure out the optimal setting of $T(d)$ and $\omega(d',d)$, which we will discuss later.

Count Expansion. In count expansion scheme, we adopt the similar idea to neighborhood based smoothing. We first propagate the contextual words to the target post to obtain pseudo counts, then estimate the language model based on the pseudo counts with maximum likelihood estimation. Instead of linearly interpolating the probabilities from different models, we interpolate the counts of words directly in this scheme. [14] suggests that count expansion tends to achieve better performance than model expansion. Formally, the ranking function can be derived as:

$$-D_{KL}(\theta_q||\theta_d) \propto \textstyle\sum_{w\in d} p_{ml}(w|\theta_q) log(1 + \frac{(1-\lambda) p_{ml}(w|\theta_{d_{exp}})}{\lambda p_{ml}(w|C)}) \tag{6}$$

where

$$p_{ml}(w|\theta_{d_{exp}}) = \frac{(1-\beta)c(w;d) + \beta \sum_{d'\in T(d)} \omega(d',d) c(w;d')}{(1-\beta)|d| + \beta \sum_{d'\in T(d)} \omega(d',d)|d'|} \tag{7}$$

where $c(w;d)$ is the count of word w in post d. The count expansion scheme is also subject to the implementation of $T(d)$ and $\omega(d',d)$. In the subsequent discussions, we will describe in detail how $T(d)$ and $\omega(d',d)$ are developed.

4.2 Utilizing Thread Contexts

As aforementioned, for both model expansion and count expansion, we need to define $T(d)$ and $\omega(d',d)$ in order to incorporate thread context. Here we discuss different possibilities for defining the two functions, each leading to a different variation of the model expansion smoothing scheme and the count expansion scheme.

Smoothing with Flat Plate Representation. The simplest way to define $\omega(d',d)$ is to give equal weight to all the posts within the same thread. This corresponds to our notion of flat plat representation (Figure 2a). Formally, $T(d)$ consists of all other posts in the thread beside d, and,

$$\omega(d', d) = \frac{1}{|T(d)|} \tag{8}$$

To further study the weighting function, we introduce two other weighting factors. They are both defined over a pair of posts within the same thread.

Inverse Structural Distance. This corresponds to the number of posts between the two posts in a certain representation of threads plus one. For example, in Figure 2a (assuming the time information is preserved in the representation), the inverse structural distance between post 1 and post 3 is 1/2. While in 2b the inverse structural distance between post 1 and 3 is 1. A large inverse structural distance usually indicates a tight relationship between two posts, if the structure is correctly represented. This is denoted by $IDist(d', d)$.

Contextual Similarity. This corresponds to the similarity between the contents of two posts. Intuitively, posts with similar word distributions would share more contexts. In practice we use cosine similarity to measure the contextual similarity. This is denoted as $Sim(d', d)$

The weighting function $\omega(d', d)$ is defined based on these two factors. We explore three different settings, corresponding to using normalized $IDist(d', d)$ and $Sim(d', d)$ each separately and using them as a combined function:

$$\omega(d', d) = \frac{IDist(d', d)Sim(d', d)}{\sum_{d'' \in T(d)} IDist(d'', d)Sim(d'', d)} \tag{9}$$

Smoothing with One-Level Tree Representation. One major problem with the flat plate representation is that it uses all the posts within a thread without really considering the internal structure of a thread. Usually, many posts within the same thread are actually about very different topics. One important observation in the thread structure is that the first post tends to be more important than all the others, as it initiates the discussion and provides the background knowledge for all the following posts. Therefore, we can use the one-level tree representation for exploring contextual information, which assigns $T(d) = \{Root(d)\}$ for any d. Figure 2b illustrates this representation. This extreme method assigns 1 to the first post and 0 to all the others in the weighting function $\omega(d', d)$. As there is only one relevant post involved, we do not need to further explore the use of structural distance or contextual similarity.

Smoothing with Timeline Representation. Smoothing with the entire thread and smoothing with only the first post are both extreme approaches. Intuitively, the context of a post is mostly captured by the contents posted before the post. Posts published after the target post may introduce off-topic information or even noise. Based on this idea, we make use of the timeline representation of the thread and design the weighting function. An example is given in 2c. We restrict $T(d) = \{d' | d' \in Thread(d) \& d' \prec d\}$, where $d' \prec d$ means d' is posted before d. Structural distance and content similarity can then be used as weighting functions.

Smoothing with Reply Tree Representation. In forums, it is possible that physically adjacent posts do not share the same topic at all. This is usually due to the problem known as topic drift. Therefore, instead of considering all contents posted before a post as the context, it makes more sense to explore the reply structure in order to get contexts. A thread starts with a single topic, and gradually grows into a multi-topic discussion. Future users tend to participate in one of the specified topics more than in the basic general topic. Finally the thread grows into a tree structure based on the reply-to relation. In this reply tree representation, the context of a post is given by the posts it replies to. These contents naturally indicate how the topic of the given post is developed. Based on this observation, we can focus on exploring the reply relation when designing the weighting functions. Particularly we restrict $T(d) = \{d'|d' \in Thread(d)\&d' \twoheadleftarrow d\}$, where $d' \twoheadleftarrow d$ means d' is on the reply path from d to the first post in the thread. Figure 2d demonstrates this representation. For $\omega(d', d)$, we apply the same combinations aforementioned. Here the structural distance corresponds to the reply distance between two posts.

5 Experiments

5.1 Data Set

A main challenge in studying forum post retrieval is the lack of a test set. We solve this problem by constructing a test set with publicly available resources.

We first obtained a full crawl of the "Computer Help" forum of CNET forum[6]. The crawl includes 29,413 threads, 25,830 users and 135,752 posts. On average, each thread has 4.6 posts and each user writes 1.13 posts. The forum is parsed so that all the metedata including the time stamp and the reply-to link are preserved. The posts are indexed so that consecutive posts within the same thread have consecutive IDs.

To generate a query set that reflects the scenario of forum search, we crawled the "Computers & Internet" category of Yahoo! Answers[7]. We then randomly select 30 question titles from the category whose words all appeared in our forum. Stopwords are filtered from the question titles, and the remaining keywords are used as queries. The average length of the test queries is 6.4 words.

We use Amazon Mechanical Turk to perform manual judgements. Particularly, we first use BM25 model [12] to retrieve the top 30 documents for each query, and ask the labelers to judge the relevance of each document. In each assignment, a labeler is presented with a question and a post, and asked to judge the post as 2, 1 or 0, representing "answers the question", "contains relevant information but not directly answers the question" and "is irrelevant" correspondingly. To guarantee the quality of the human labelers, we require the participants to have a task approval rate of more than 95%. During the judging

[6] http://forums.cnet.com/computer-help-forum/

[7] http://answers.yahoo.com/dir/index?sid=396545660

process, we also monitor the behaviors of labelers and reject abnormal submissions, e.g. time usage less than 5 seconds. In total, 73% submitted assignments are accepted and the average time usage for each assignment is 17 seconds.

Five labelers are employed to judge each document. The final judgement is given by majority voting. In the case of tie, the document is judged as "contains relevant information but not directly answers the question". The average human agreement rate is 0.63. There are on average 7.8 documents judged as 2, 15.3 as 1 and 6.9 as 0. As we can see, the labelers tend to judge posts as 1 – "contains relevant information but not directly answers the question". In order to guarantee the quality of retrieval, we take level 2 as relevant, level 1 and 0 are both irrelevant.

We evaluate the performance of all the methods by doing re-ranking on the test set. This is because we are experimenting on different retrieval models and incremental pooling is too expensive. The metrics we use for evaluation are Mean Average Precision (MAP), Precision at 1, 5 and 10 (P@1, P@5 and P@10). All the reported results are based on 5-fold cross validation w.r.t. the MAP measure.

5.2 Experiment Results

Existing Retrieval Models. Since forum post retrieval is a new retrieval task, we first compare the performance of different existing retrieval models for this task. We see that BM25 model and language model achieve comparable performance. Jelinek Mercer smoothing gives better performance than Dirichlet smoothing. This is consistent with [19]'s finding that Jelinek Mercer smoothing outperforms Dirichlet smoothing on long queries due to better modeling of common words in the query.

Table 1. Performance of Existing Retrieval Models

	MAP	P@1	P@5	P@10
BM25	0.445	0.467	0.373	0.343
LM+DIR	0.415	0.367	0.387	0.357
LM+JEL	0.457	0.467	0.373	0.357
CBDM	0.434	0.433	0.347	0.317
CBDM2	0.487	0.433	0.407	0.343
DELM	**0.489**	0.467	0.367	0.353

We also see that CBDM does not actually outperform language model, while DELM improves the performance significantly. A slightly modified version of CBDM, namely CBDM2, rewrites the CBDM formula to achieve the IDF effect. CBDM2 achieves a slightly lower performance than DELM. This is also in accordance with [14]'s findings. These methods serve as the baselines for studying the effectiveness of our proposed new smoothing methods.

Smoothing with Model Expansion. Table 2 shows the performance of the model expansion scheme w.r.t. all the combination of ways to utilize the thread context. For simplicity, we denote model expansion scheme as "ME" and count expansion scheme as "CE" in the following. We also denote flat plate representation, one-level tree representation, timeline representation and reply tree representation of threads as "FL", "ON", "TI" and "RE" correspondingly. The four weighting formula – equal weight, inverse structural distance, contextual similarity and the combination of inverse structural distance and contextual similarity, are denoted as "EQ", "DS", "SI" and "DSSI" respectively.

Table 2. Model Expansion

	MAP	P@1	P@5	P@10
ME+FL+EQ	0.489	0.467	0.380	0.350
ME+FL+DS	0.494	0.400	0.393	0.373
ME+FL+SI	0.492	0.433	0.400	0.367
ME+FL+DSSI	0.499	0.433	0.413	0.363
ME+ON+EQ	0.504	0.467	0.393	0.360
ME+TI+EQ	0.492	0.467	0.407	0.350
ME+TI+DS	0.496	0.467	0.400	0.360
ME+TI+SI	0.506	0.467	0.427	0.363
ME+TI+DSSI	0.508	0.567	0.400	0.367
ME+RE+EQ	0.503	0.467	0.393	0.367
ME+RE+DS	0.508	0.500	0.420	0.363
ME+RE+SI	0.513	0.500	0.413	0.373
ME+RE+DSSI	**0.515**	0.533	0.420	0.363

Table 3. Count Expansion

	MAP	P@1	P@5	P@10
CE+FL+EQ	0.493	0.500	0.373	0.363
CE+FL+DS	0.494	0.500	0.387	0.367
CE+FL+SI	0.493	0.500	0.380	0.367
CE+FL+DSSI	0.503	0.433	0.393	0.370
CE+ON+EQ	0.511	0.433	0.393	0.363
CE+TI+EQ	0.507	0.500	0.387	0.357
CE+TI+DS	0.516	0.467	0.407	0.367
CE+TI+SI	0.498	0.467	0.387	0.370
CE+TI+DSSI	0.507	0.500	0.387	0.370
CE+RE+EQ	0.509	0.467	0.360	0.377
CE+RE+DS	0.515	0.500	0.360	0.380
CE+RE+SI	0.517	0.500	0.373	0.380
CE+RE+DSSI	**0.523**	0.533	0.380	0.380

In the results we see that almost all of the combinations in ME outperform the existing retrieval methods. This verifies our hypothesis that threads are natural clusters of posts. We also see that the best performance is achieved by ME+RE+DSSI. A deeper analysis suggests that using RE representation gives us better performance than any other representations. This shows the importance of utilizing the reply structure of forum threads, as they provide "cleaner" context for posts. Meanwhile, in the experiments, DSSI consistently performs better than other weighting techniques, and it is also demonstrated that both structural distance and contextual similarity contribute to the improvement of accuracy. Another interesting finding is that smoothing with only the first post achieves fairly good performance. Unlike smoothing with other representations of threads, the first post is always guaranteed to be relevant to the target post. Therefore, we see in the results that ME+ON+EQ outperforms almost all the other representations with equal weighting function. However, after structural distance and contextual similarity are used for weighting functions, the other representations can outperform or at least catch up with the performance of ME+ON+EQ.

Smoothing with Count Expansion. Table 3 shows the performance of the count expansion scheme w.r.t. all the combination of ways to utilize thread context. Again, we see that all the combinations in CE outperform the existing retrieval methods, further confirming the effectiveness of the proposed new smoothing methods. . Compared with ME, we can see that the optimal performance of CE, which is achieved by CE+RE+DSSI, is slightly better. This is in accordance with the previous findings in [14] that count expansion tends to be superior compared with model expansion. Besides, we see almost all of the findings in the previous subsection are also verified in the count expansion scheme. Therefore, we are able to conclude that thread structure in forums provides highly useful contextual information for posts. By appropriately exploiting the thread structure, we are able to significantly improve the state-of-the-art retrieval methods in forum post retrieval.

Statistical Significance. We perform t-test over all the runs in our experiment. The runs that outperform all the existing retrieval methods significantly (p-value<0.05) are shown in bold font in Table 2 and Table 3. This indicates that (1) modeling threads with reply tree representation is the most important factor in improving the performance of forum post retrieval, and (2) the use of context similarity and reply distance for expansion is critical in achieving a significant improvement.

6 Conclusions

This paper studies the problem of forum post retrieval. The contributions of this paper are as follows. First, to the best of our knowledge, this is the first systematic study of the problem of forum post retrieval by exploiting the *raw* thread structures of forums. We constructed a test set for forum post retrieval; while our data set is small, it is possible to differentiate different retrieval methods with statistically significant results. We propose and explore two new smoothing schemes to exploit the thread structure in order to improve smoothing of language models. We also propose and study different ways of utilizing the contextual information within the smoothing schemes. Finally, extensive experiments are carried out on the test data we constructed and it is demonstrated that our proposed smoothing schemes can improve the accuracy of forum post retrieval significantly.

As for future work, we plan to explore several directions to improve forum post retrieval. First, we plan to further explore the problem of forum post retrieval on a larger scale of data. The current evaluation set is relatively small due to the limitation of resources. We plan to study how to scale up the evaluation without incurring too much cost. Meanwhile, we also plan to take into consideration the static ranking of forum posts to further optimize retrieval performance.

7 Acknowledgments

This paper is based upon work supported in part by MIAS, the Multimodal Information Access and Synthesis center at UIUC, part of CCICADA, a DHS Center of Excellence, and by the National Science Foundation under grants IIS-0713581 and CNS-0834709.

References

1. http://www.boardtracker.com/cgi-bin/about.pl?page=1.
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, New York, NY, USA, 2008. ACM.
3. D. Hiemstra. Statistical language models for intelligent XML retrieval. In *Intelligent Search on XML Data*, pages 107–118, 2003.
4. D. Hiemstra and W. Kraaij. Twenty-one at trec-7: Ad-hoc and cross-language track. In *TREC '99*, pages 227–238, 1999.
5. L. Hong and B. D. Davison. A classification-based approach to question answering in discussion boards. In *SIGIR '09*, 2009.
6. J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR'01*, pages 111–119, Sept 2001.
7. C. Lin, J.-M. Yang, R. Cai, X.-J. Wang, and W. Wang. Simultaneously modeling semantics and structure of threaded discussions: a sparse coding approach and its applications. In *SIGIR '09*, pages 131–138, New York, NY, USA, 2009. ACM.
8. X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *SIGIR '04*, pages 186–193, New York, NY, USA, 2004. ACM Press.
9. D. R. H. Miller, T. Leek, and R. M. Schwartz. BBN at trec7: Using hidden markov models for information retrieval. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 80–89, 1998.
10. P. Ogilvie and J. Callan. Hierarchical language models for xml component retrieval. In *Proceedings of INEX Workshop*.
11. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281, New York, NY, USA, 1998. ACM Press.
12. S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*, pages 232–241, 1994.
13. J. Seo, W. B. Croft, and D. A. Smith. Online community search using thread structure. In *CIKM '09*, pages 1907–1910, New York, NY, USA, 2009. ACM.
14. T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *HLT-NAACL '06*, pages 407–414, Morristown, NJ, USA, 2006. Association for Computational Linguistics.
15. W. Weerkamp, K. Balog, and M. D. Rijke. Using contextual information to improve search in email archives. In *ECIR '09*, 2009.
16. G. Xu and W.-Y. Ma. Building implicit links from content for forum search. In *SIGIR '06*, pages 300–307, New York, NY, USA, 2006. ACM.
17. C. Zhai. *Statistical Language Models for Information Retrieval*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2008.
18. C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*, pages 403–410, New York, NY, USA, 2001. ACM Press.
19. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, pages 334–342, New York, NY, USA, 2001. ACM Press.
20. C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *SIGIR '02*, 2002.