Lee JT, Yang MC, Rim HC. Discovering high-quality threaded discussions in online forums. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 29(3): 519–531 May 2014. DOI 10.1007/s11390-014-1446-5

Discovering High-Quality Threaded Discussions in Online Forums

Jung-Tae Lee, Min-Chul Yang, and Hae-Chang Rim*, *Member, ACM*

Department of Computer and Radio Communications Engineering, Korea University, Seoul 136-713, Korea

E-mail: {jtlee, mcyang, rim}@nlp.korea.ac.kr

Received September 25, 2013; revised December 22, 2013.

Abstract Archives of threaded discussions generated by users in online forums and discussion boards contain valuable knowledge on various topics. However, not all threads are useful because of deliberate abuses, such as trolling and flaming, that are commonly observed in online conversations. The existence of various users with different levels of expertise also makes it difficult to assume that every discussion thread stored online contains high-quality contents. Although finding high-quality threads automatically can help both users and search engines sift through a huge amount of thread archives and make use of these potentially useful resources effectively, no previous work to our knowledge has performed a study on such task. In this paper, we propose an automatic method for distinguishing high-quality threads from low-quality ones in online discussion sites. We first suggest four different artificial measures for inducing overall quality of a thread based on ratings of its posts. We then propose two tasks involving prediction of thread quality without using post rating information. We adopt a popular machine learning framework to solve the two prediction tasks. Experimental results on a real world forum archive demonstrate that our method can significantly improve the prediction performance across all four measures of thread quality on both tasks. We also compare how different types of features derived from various aspects of threads contribute to the overall performance and investigate key features that play a crucial role in discovering high-quality threads in online discussion sites.

Keywords online forum, discussion board, thread quality

1 Introduction

The Web has now become a place where you not only look for information but freely interact with other people that share similar interests or concerns. An online forum, sometimes referred to as a discussion board, is an online site or an area on a website where users of the site can freely hold discussions with others about any topic. A forum differs from online chatting in that user conversations are archived for others to read in the form of threaded discussions. A thread refers to a series of user-generated message posts built over time on a topic. Over the years, many forum sites have been able to accumulate tremendous volumes of discussion threads on various topics. Those forum archives potentially have huge amounts of collective knowledge, such as answers to questions or real opinions and responses from groups of individuals that many information seekers would be interested in. For major search engine companies that maintain their own local search index to facilitate fast information retrieval, such resources are found attractive to store.

However, the quality of forum threads being created by real users varies widely from informative, interesting conversations to useless, extraneous contents. This is in many cases due to abusive behavior, such as trolling or flaming, with the intent of disrupting an on-going discussion. Moreover, due to the existence of forum users with varied backgrounds and skills, we cannot assume that every thread stored in online forums is of high quality. Although some forums assign moderators who are given special privileges to delete or edit threads in order to keep discussions appropriate, there is always a limit as to how many threads people can moderate manually. Therefore, automatically finding high-quality threads in online discussion sites is desirable and valuable. It can benefit not only online forum sites in serving more useful contents to their end-users but search engines in maintaining more efficient index and enhancing search quality.

Regular Paper

This research was partially supported by the Ministry of Knowledge Economy (MKE), Korea, and Microsoft Research through the IT/SW Creative Research Program supervised by the National IT Industry Promotion Agency (NIPA) of Korea under Grant No. NIPA-2012-H0503-12-1012, and the Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning of Korea under Grant No. NRF-2012M3C4A7033344.

Part of this work was done while the first author was an intern at Microsoft Research Asia, Beijing.

*Corresponding Author

Processing forum archives in thread-level is also essential in that a discussion in a thread logically builds a document unit on a particular topic or a goal. From user perspective, the first thing that a user sees when the user visits a forum site is its index page listing the titles of archived threads. When the user chooses one to view, the corresponding thread is displayed as a whole. Viewing a whole thread enables the user to not only focus on a particular message in the thread but locate influential contributions by backtracking to its previous messages. Many previous studies on online forums also recognize a thread as a unit of information to be retrieved (e.g., [1-3]).

Although forum archives have recently gained considerable attention as useful online resources, only little attempt has been made to date to automatically distinguish high-quality threads in the archives. To the best of our knowledge, there is no published work in literature that directly addresses the problem of mining high-quality threads. There are some online forums that simply use a single attribute of a thread, such as the total number of message posts in each thread, to serve "hot" or popular threads in their index pages. However, the popularity of a thread reflected from its post count does not necessarily reflect or guarantee the quality of the discussion. Besides post counts, forum archives exhibit a variety of other metadata, but there is no reported analysis of them with regard to thread quality.

In this paper, we start out by defining a high-quality thread broadly as one that has contents of which most of its readers would highly evaluate or recommend to others. Although it is hard to find any online forum that enables users to rate a thread as a whole, there are a few forum sites that have user moderation systems which enable users to collaboratively rate posts in a thread. We assume that forum archives with post ratings derived from readers' feedback can provide an alternative channel for learning important relationships and correlations between forum threads and their quality. As a test case, we focus on Slashdot, a popular open forum site that represents a definitive example of user moderation system. We intend to investigate the following research questions:

1) How can we measure the overall quality of a thread inferred from the ratings of its posts?

2) Can we effectively discover high-quality threads without actually analyzing the content of threads?

3) Which attributes of threads are most associated with thread quality? What does a high-quality thread look like in terms of shape or structure?

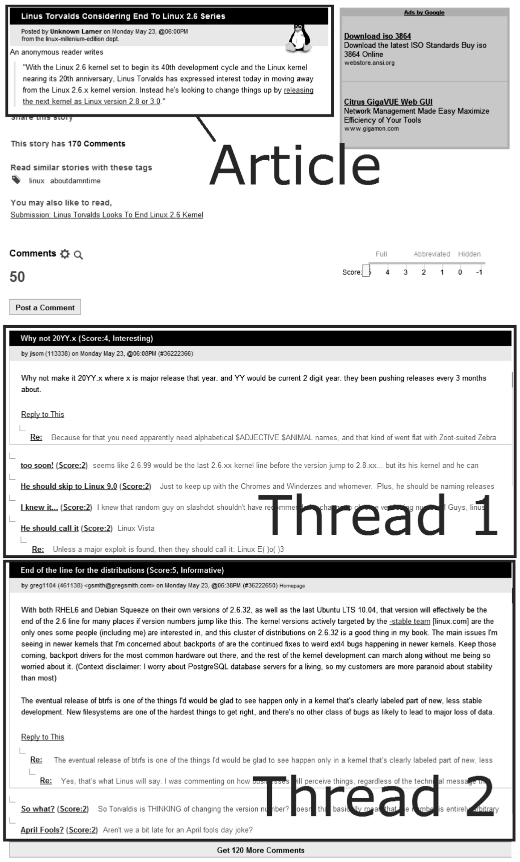
In particular, we present several artificial metrics for measuring the overall quality of a thread based on the aggregation of ratings of individual posts in the thread. We then propose two novel prediction task scenarios involving thread quality. The first task aims at ranking a set of threads related to a particular topic according to their quality. The latter is, given a pair of threads initiated at a same time on a same topic, to select which thread is likely to be of higher quality than the other. Post rating information is not to be provided at the prediction stage. We address both tasks as ranking problems and adopt a well-known learning-to-rank approach to perform the predictions. We investigate several features extractable from discussion threads without attempting to understand the language of thread contents. Our method is practically applicable to any online forum data, since we do not use any service or domain-specific features. Experimental results demonstrate that our method of utilizing multiple features extracted from different aspects of threads shows significant improvement over baseline approaches across all thread quality metrics for both prediction tasks. We also perform a comparative study of individual features to discuss key features in discovering high-quality threads in online forums.

It is important to note that the purpose of this research is to help users select high-quality forum threads to read in a *browsing* scenario not in a *search* scenario. The main difference between the two is that there is no query involved in the browsing scenario while in forum thread retrieval and ranking there is always a query involved. Browsing scenario is also important as it is similar to the situation when a search engine crawler discovers a forum archive and needs to decide which forum thread to index or not before it serves indexed threads to a particular query. Therefore, we provide no comparison with existing online forum search algorithms here, because we focus on predicting the quality of forum contents not retrieving relevant forum contents with regard to a particular query. We should also note that we attempt to predict the quality of a thread not when only one or few posts of the thread are given but when ratings of posts are not available, as is common in most online forums.

The rest of this paper is organized as follows. Section 2 introduces Slashdot. Section 3 presents several metrics for calculating thread quality based on post ratings. Section 4 explains how we approach the thread quality prediction problem. Section 5 reports experimental results. Section 6 discusses related work. Section 7 finally concludes the paper.

2 Slashdot Forum

Slashdot is a popular forum site where users hold discussions on a number of current affairs and news stories related to technology and science. Summaries (and links) of interesting news articles are submitted by users, and each article becomes the topic of discussions among users. Users can write not only comments about the particular article but replies to other users' comments. A reply to a comment is placed underneath that comment in a nested structure. As a result, each article page contains one or more threaded discussions in its comments section attached to the article. Fig.1 illustrates this. Discussion threads in an article page may be focused on different subtopics but are derived basically from the topic of the article.



Linus Torvalds Considering End To Linux 2.6 Series

Posted by **Unknown Lamer** on Monday May 23, @06:00PM
from the linux-millenium-edition dept.

An anonymous reader writes

> "With the Linux 2.6 kernel set to begin its 40th development cycle and the Linux kernel nearing its 20th anniversary, Linus Torvalds has expressed interest today in moving away from the Linux 2.6.x kernel version. Instead he's looking to change things up by releasing the next kernel as Linux version 2.8 or 3.0."

Share this story

This story has **170 Comments**

Read similar stories with these tags

linux aboutdamntime

You may also like to read,
Submission: Linus Torvalds Looks To End Linux 2.6 Kernel

Ads by Google

Download iso 3864
Download the latest ISO Standards Buy iso 3864 Online
webstore.ansi.org

Citrus GigaVUE Web GUI
Network Management Made Easy Maximize Efficiency of Your Tools
www.gigamon.com

Comments

50

Full Abbreviated Hidden
Score: 5 4 3 2 1 0 -1

Post a Comment

Why not 20YY.x (Score:4, Interesting)

by jisom (113338) on Monday May 23, @06:08PM (#36222366)

Why not make it 20YY.x where x is major release that year. and YY would be current 2 digit year. they been pushing releases every 3 months about.

Reply to This

Re: Because for that you need apparently need alphabetical $ADJECTIVE.$ANIMAL names, and that kind of went flat with Zoot-suited Zebra

too soon! (Score:2) seems like 2.6.99 would be the last 2.6.xx kernel line before the version jump to 2.8.xx... but its his kernel and he can

He should skip to Linux 9.0 (Score:2) Just to keep up with the Chromes and Winderzes and whomever. Plus, he should be naming releases

I knew it... (Score:2) I knew that random guy on slashdot shouldn't have recommended the change to choose versioning numbers! Guys, linus

He should call it (Score:2) Linux Vista

Re: Unless a major exploit is found, then they should call it: Linux E( )o( )3

End of the line for the distributions (Score:5, Informative)

by greg1104 (461138) <gsmith@gregsmith.com> on Monday May 23, @06:38PM (#36222650) Homepage

With both RHEL6 and Debian Squeeze on their own versions of 2.6.32, as well as the last Ubuntu LTS 10.04, that version will effectively be the end of the 2.6 line for many places if version numbers jump like this. The kernel versions actively targeted by the -stable team [linux.com] are the only ones some people (including me) are interested in, and this cluster of distributions on 2.6.32 is a good thing in my book. The main issues I'm seeing in newer kernels that I'm concerned about backports of are the continued fixes to weird ext4 bugs happening in newer kernels. Keep those coming, backport drivers for the most common hardware out there, and the rest of the kernel development can march along without me being so worried about it. (Context disclaimer: I worry about PostgreSQL database servers for a living, so my customers are more paranoid about stability than most)

The eventual release of btrfs is one of the things I'd would be glad to see happen only in a kernel that's clearly labeled part of new, less stable development. New filesystems are one of the hardest things to get right, and there's no other class of bugs as likely to lead to major loss of data.

Reply to This

Re: The eventual release of btrfs is one of the things I'd would be glad to see happen only in a kernel that's clearly labeled part of new, less

Re: Yes, that's what Linus will say. I was commenting on how businesses will perceive things, regardless of the technical message th

So what? (Score:2) So Torvaldis is THINKING of changing the version number? Doesn't that basically mean that the number is entirely arbitrary

April Fools? (Score:2) Aren't we a bit late for an April fools day joke?

Get 120 More Comments

Fig.1. Sample article page in Slashdot.

What makes the site unique is that individual comments have ratings which have been accumulated from multiple readers. Slashdot has a user-based moderation system that many forums do not have. Users who are eligible to become moderators are randomly chosen and given a limited number of comment posts to moderate for a certain period of time. Users moderate each comment post by selecting a word from a list that appears next to the comment containing several words, such as "Informative" and "Off-Topic." Positive words (e.g., "Informative") will increase the overall rating of the comment post by a single point, and negative words (e.g., "Off-Topic") will reduce the rating by a single point. All comments in Slashdot have a score in the range of -1 to 5 points. Refer to Slashdot FAQ page[1] for detail.

The reason why Slashdot has this type of user moderation system is to maintain the site as readable for as many users as possible. When users are viewing the site via their browsers, they can alter a threshold widget so that only the contents meeting or exceeding a certain threshold would be viewed. For example, if a user chooses 3 as the threshold, the user will only see posts with rating of 3 or higher; posts with ratings below 3 will be either abbreviated or become hidden in the browser. We believe that the scores of individual posts which have been accumulated from multiple moderators in Slashdot are valuable resources to learn how a large community of readers perceives the quality of contents generated by others in forums.

3 Measuring Thread Quality

It is rational to argue that a discussion thread is of high quality if the discussion possesses content that many people have considered useful or interesting. Although a rating of a post in Slashdot may be a subjective score, it virtually reflects what most readers of the post felt. Thus we define a high-quality thread in an online forum broadly as follows.

Definition 1. *A high-quality thread in an online forum is a thread that possesses posts in which the majority of readers have highly rated or recommended.*

Formally, a forum is structured as a pair $\langle U, C\rangle$ where $U = \{u_1, u_2, \ldots, u_{n_u}\}$ is a set of forum users, and $C = \{c_1, c_2, \ldots, c_{n_c}\}$ is a set of subforums (or categories), each covering a specific topic related to the forum's theme. A subforum $c_i \in C$ is composed of topically relevant threads $\{t_1, t_2, \ldots, t_{n_t}\}$. A thread t_j is a set of user-created posts $\{p_1, p_2, \ldots, p_{n_p}\}$. A post p_k is a message created by a user at a particular time.

The concept of user moderation introduced in Section 2 can also be formalized as a triple (u, p, v) where $u \in U$ is a forum user, $p \in t_i$ is a post in a thread that is to be moderated, and $v \in \{+1, -1\}$, in which $+1$ means that a user u has given a positive vote to a post p and -1 means vice versa. Let $P^+ = \{(u, p, v) : v = +1\}$ be a set of triples with positive votes and $P^- = \{(u, p, v) : v = -1\}$ be a set of triples with negative votes. We can define a rating function $R(p)$ that specifies the difference between positive and negative voters of a post p, i.e., $|\{u : (u, p, v) \in P^+\}| - |\{u : (u, p, v) \in P^-\}|$ where $|\{\cdot\}|$ is the size of a set.

The problem of measuring thread quality based on user moderation can be defined as follows.

[1] http://slashdot.org/faq, Jan. 2014.

Thread Quality Measurement Problem. Given a thread t with n_p rated posts $\{p_1, p_2, \ldots, p_{n_p}\}$ and a rating function $R(p_i)$ that outputs the rating of a post $p_i \in t$, measure the overall quality of the thread $S(t)$.

This problem is however a subjective task, because the judgments of individuals may vary depending on their preference. For example, some users would prefer to read threads that contain many highly rated posts although such threads may be very long in length. In contrast, some would prefer to read concise threads with higher ratio of highly rated posts. Some users might prefer somewhere in between. Therefore, we propose four different artificial metrics for measuring the overall quality of a thread based on post ratings.

3.1 Rating Ratio

The intuition behind our first artificial metric is that the higher the ratio of worth reading posts in a thread, the more likely users will expect higher thread quality. Let Δv be a parameter that specifies the margin of separation in terms of number of votes between P^+ and P^-. We consider a post p in a thread to be worth reading if and only if the post has obtained a rating of at least Δv points from users' votes, i.e., $R(p) \geqslant \Delta v$. In this paper, we set $\Delta v = 3$, because Slashdot recommends users to view posts with rating of 3 or higher.

We now define the rating ratio of the thread t as the ratio of the count of worth reading posts to that of total posts in t, as follows:

$$S_{\text{Ratio}}(t) = |\hat{t}|/|t|, \tag{1}$$

where $|t|$ refers to the size of t, and $\hat{t}$ refers to the set of posts in t that are worth reading, i.e., $\hat{t} = \{p \in t : R(p) \geqslant \Delta v\}$.

3.2 Absolute Rating

The previous rating ratio metric captures the key aspect of thread quality but does not consider the distinction between possibly worth reading and completely worth reading, e.g., post rating of 3 versus 5 in case of Slashdot.

The intuition of the second metric is that the more highly rated posts a thread has, the more likely users will decide to read it. We define the absolute rating of a thread t by the summation of ratings of worth reading posts in t, as follows:

$$S_{\text{Abs}}(t) = \sum_{\forall p_i \in \hat{t}} R(p_i). \tag{2}$$

Note that absolute rating ignores posts of low quality in t, i.e., $\{p \in t : R(p) < \Delta v\}$, in the summation. The reason why we only consider worth reading posts rather than all posts in a thread is to prevent extremely long threads with many low quality posts from receiving higher scores than comparably shorter threads that may have more high-quality posts than the longer threads.

3.3 Average Rating

The drawback of the previous absolute rating metric is that threads with more posts would still have more chances to get higher scores. Therefore, we present another way to define thread quality as the average rating of worth reading posts in the thread, namely average rating, as follows:

$$S_{\text{Avg}}(t) = \frac{1}{|\hat{t}|} \sum_{\forall p_i \in \hat{t}} R(p_i). \tag{3}$$

For simplicity, we assume that $S_{\text{Avg}}(t) = 0$ if $|\hat{t}| = 0$.

Note that this measure emphasizes the average post quality of a thread and does not favor threads with more posts like the absolute rating metric.

3.4 Bayesian Average Rating

Assume there is a thread with only one post rated 5 points and another thread with hundreds of posts rated 5 points each. It is reasonable to argue that the latter is more worth reading, but the average rating measure would fail to distinguish the difference between the two threads.

The intuition behind our last artificial metric is that if there is only a small number of high-quality posts in a thread, their ratings should count less than when there are many high-quality posts. A common way to do this is to use the Bayesian average. The Bayesian average rating of a thread t can be defined as follows:

$$S_{\text{BAvg}}(t) = \frac{1}{C + |\hat{t}|} \times \Big(Cm + \sum_{\forall p_i \in \hat{t}} R(p_i)\Big), \tag{4}$$

where C is the average number of posts in threads, and m is the average score of threads. Under this criterion, the score of a thread will become closer to the average score of threads when it has fewer posts. For simplicity, we also assume that $S_{\text{BAvg}}(t) = 0$ if $|\hat{t}| = 0$.

4 Predicting Thread Quality

We now focus on the problem of discovering high-quality threads by predicting their quality. Such a problem is important, because most online forums in real world do not maintain ratings for individual contents posted on their sites systematically as Slashdot does. One straightforward solution is to assign a given thread an either "good" or "bad" label. The main drawback is that it is virtually impossible to set an absolute, clear definition of good or bad threads. Another disadvantage of the classification approach is that it is inconvenient to control the number of high-quality threads to be extracted from given forum archive according to the requirements of real world applications, such as indexing and retrieval. Thus, it is more appropriate to view the problem of finding high-quality threads as a ranking task.

Recall from Section 3 that a typical online forum is logically organized into a finite set of subforums, each covering a topic. In Slashdot's case, each news article page can be regarded as a subforum, since threaded discussions arise on the topic of the news article. It is reasonable to rank threads that cover a similar topic rather than threads from different topics, because users with topically different interest may have varied preference towards the quality of discussions or the degree of participation. We now formalize the problem of thread ranking as follows.

Thread Ranking Problem. Let $S(t)$ be one of the thread quality metrics presented in Section 3. Given a set of threads $\{t_1, t_2, \ldots\} \in c$ originated from a same topic, the task is to rank the threads such that t_i has a higher ranking than t_j if and only if $S(t_i) > S(t_j)$, without referring to the rating of each post $p \in t$.

This ranking problem is certainly a real and interesting task, especially to Web applications. Being able to rank threads by their assessed quality would be an interesting feature for online forums that could help users find useful contents in a more convenient way.

However, there is an issue in terms of evaluation. Discussion threads in Slashdot, which we will use for testing our method, are created in different time periods. Thus, there is no guarantee that all threads in the evaluation data have had equal chance of exposure to users. Therefore there is a possibility that some threads' posts have low ratings, not because they deserve it, but because only a few users have read and participated to moderate them. Threads that cover unpopular topics or have been created very recently may be examples of such cases. To alleviate the issue, we propose another prediction problem that considers not only the topic of threads but the creation time of the threads.

Pairwise Thread Selection Problem. Let x_t be a specific time in which a thread t is firstly created and Δx be a parameter that specifies the margin of separation in terms of minutes between two threads. Given a pair of online forum threads $\{\langle t_1, t_2\rangle : t_1 \in c, t_2 \in c\}$ from a same subforum where $|x_{t_1} - x_{t_2}| < \Delta x$, the task is to select a thread t_i such that $S(t_i) > S(t_j)$, without referring to the rating of each post $p \in t$.

Although the second problem is less applicable to real world than the first, we can assure a more fair comparison between competing threads in evaluations if we consider both the topic and the time of creation of threads. This is based on the assumption that if threads share the same topic and occur at the same time, they would have more chance to be displayed near each other in users' browsers, which leads to an equal chance of exposure to users for moderation.

Our study aims at presenting a framework that works well for both problems. We consider both thread ranking and pairwise thread selection problems as ranking tasks and adopt one of the best-known learning-to-rank approaches based on the support vector machine classifier, known as Ranking SVMs[4] to learn a ranking function for threads. Here we introduce a range of features we use for predicting thread quality. We only focus on features extractable without analyzing the content of each thread using natural language processing techniques. Note that we try not to use any features that are specific to Slashdot in order to let our method be applicable to other generic forum archives.

4.1 Features from Surface of Thread

• *Number of Posts in the Thread.* This is typically the only numerical hint that users can refer to when deciding which thread to read from a list of threads in an index page of forums. It indicates the degree of participation in a particular thread.

• *Number of Unique Users in the Thread.* The amount of users participating in a discussion may be proportional to the discussion's quality.

• *Length of the Thread.* Length has been proven to be a simple but very effective feature for determining the quality of user-generated texts, such as answers to questions[5-6]. In this paper, we consider the total length of the whole thread, the length of the initial post, and the total and average lengths of the reply posts as length features.

• *Average Number of Posts per User.* This number indicates whether posts haven been generated evenly by all users or most generated by a few of the users.

• *Whether the Thread Is Initiated by a Registered User.* In many online forums, a user can decide whether to log-on or not before they create some content. Anonymous users tend to post useless posts.

• *Number of Unique Registered Users.* We consider both the total number of users who had logged on before generating posts and the ratio of them to all users in the thread.

• *Number of Unique Revisiting Users.* Users posting more than one message in a thread may be a sign that they want to continue the on-going discussion.

- *Thread Depth.* If a thread is considered as a tree structure rooted from its initial post, the length of the path from the initial post to the deepest reply post would be the thread's depth. This number indicates the growth of a discussion. We normalize the value of this feature by the number of posts in the thread.

- *Thread Width.* This feature corresponds to the largest number of posts in one level within a thread. It indicates whether the thread carries an issue that many people choose to make comments simultaneously. We also normalize the value of this feature by the number of posts in the thread.

- *Average Thread Width.* This feature refers to the ratio of thread width to thread depth. It roughly indicates whether the arrangement of posts in the thread is spread horizontally or vertically.

- *Number of Quotes.* Some forums have a unique feature that lets users to quote some text from a previous post. The frequency of quotes may indicate whether the discussion is more focused on certain points. We use the total number of quotes appeared in a thread and the average number of quotes per post as features.

- *Number of Posts by Reply-Inducing Users.* While analyzing the Slashdot users, we have observed that the average rating of posts a user creates is positively correlated to the average number of replies the user receives per post. Similar pattern of correlation is also reported by Gómez *et al.*[7] If there are many users in a thread who have received at least one reply every time they write, we may assume that there would be many posts by authoritative users. We count the total number of posts written by such users and the ratio of the posts to the total number of posts as features.

4.2 Features Reflecting Time Duration

- *Duration of the Thread.* The continuance of a discussion in terms of time may indicate continuous interest of users toward a topic. We calculate the total duration by the length of the time between the created time of the initial post and the last reply post.

- *Latency of Replies.* This refers to the time elapsed from a post receives a reply. We consider the total latency time and the average latency time for individual post pairs in reply-to relations.

- *Duration of Replies to the Initial Post.* This refers to the time passed between the creation of an initial post and the creation of each reply post. Many forum sites choose to show posts simply in chronological order rather than explicitly displaying reply-to relations between posts. We consider the total duration and the average duration for individual reply posts.

4.3 Features Indicating Authority of Initiator

Many forum sites display the name of the user who initiates a thread (i.e., who writes the first post of the thread requesting a discussion) in their index pages. Readers may not notice anything directly from the name when viewing the index page, but each user's activity log obtained from the site's archive may contain some useful indicators that would reflect the authority of the user and, moreover, the quality of a new thread the user initiates.

- *Number of Initiated Threads.* This refers to the number of threads in the forum archive that have been started by the initiator.

- *Number of Participated Threads.* This refers to the number of threads that the initiator did not initiate but participated by writing reply posts.

- *Number of Reply Posts Written.* We count the total number of the reply posts written by the initiator of the thread as a feature.

- *Number of Reply Posts Received.* From the Slashdot data, we have observed that the number of reply posts a user received has positive correlation to the average ratings of the posts the user generated. This observation indicates that high-quality posts induce more reply posts. Here we consider the total number and the average number of reply posts the initiator received per post as features.

- *Number of Revisiting Threads.* This refers to the number of threads in which the initiator wrote more than one post. We consider both the total number and the ratio of such threads to all threads the user participated.

- *Number of Participants Induced.* This refers to the number of users who joined discussion threads that the initiator started. The total number and the average number of participants per thread started by the initiator are used as features.

- *Number of Revisiting Users Induced.* This is similar to the previous one except that only the participants who wrote more than one post in a thread are counted.

- *Number of Co-Participants.* This refers to the number of other participants besides the initiator in those threads that the initiator participated. It indicates whether the initiator likes to join discussions with many users or not. Here we use the total number and the average number of co-participants per thread the initiator participated.

4.4 Features on Authority of Participants

Not only the initiator who started a thread but users that participated in the discussion can be another clue for predicting the overall quality of that thread. We extract all the same activity features above also for all the participants. Because there can be multiple participants in a thread, we consider both the total number and the average number per participant as features. For example, when calculating the number of initiated threads, we consider both the number of all threads initiated by all the participants and the average number of threads initiated per participant.

5 Experiment

5.1 Setup

During the second half of the year 2010, we have crawled the complete archive of Slashdot website comprised of news articles as well as their subsequent user comment threads. There have been cases where some comments in the crawled archive refer to parent comments that had not been successfully collected. There were also cases in which comments did not have some important information, such as timestamps, attached to them. After removing those comments, we were finally able to compile a new Slashdot dataset consisting of 444 912 comment posts in 2 878 news article pages. This dataset has not been released to public yet, but we have plan to share it upon request for research purposes only, if appropriate. Table 1 shows the statistics of our final Slashdot dataset. Note that the average rating of posts in our Slashdot corpus is 2.12. Although Slashdot recommends its visitors to view posts with rating of 3 or higher, only 26% of the entire posts were revealed to have rating of at least 3 in our dataset. This implies that discovering high-quality threads with worth reading posts is a non-trivial task.

Table 1. Summary Statistics of Our Slashdot Dataset

Total number of news articles	2 878.00
Total number of threads attached to articles	29 384.00
Total number of posts attached to articles	444 912.00
Average number of posts per thread	15.14
Average rating of posts	2.12
Total number of users (registered)	42 842.00

We use the SVM-Light software[8] for training Ranking SVM models with linear kernels. All feature values are normalized in the range from 0 to 1 before training and testing. For the thread ranking task, we report the performance of ranked thread lists in terms of normalized discounted cumulative gain (NDCG)[9]. This measure gives a high score to a ranked list where high-quality threads are ranked higher than low-quality ones. For the pairwise thread selection task, we report the error rate of preference pairs[4], which is simply the ratio of the number of mistakenly ranked pairs to the number of all pairs in the evaluation set. All experimental results are obtained through 10-fold cross validation. We use two-tailed, paired t-tests for significance tests. We set the p-value to 0.05, which means that we consider the difference in effectiveness of two different methods with p-value less than 0.05 as statistically significant.

In order to test whether our method can improve performance in both tasks, we compare our approach with three representative baseline approaches. First, considering from the standpoint of a user or a search engine crawler about to choose which thread to read or index respectively, it would be ideal if the quality indicator is observable directly from forum index pages where choices of threads are listed. Thus, we choose the number of posts in a thread (denoted as #Posts) as our first baseline, since that information usually appears next to threads in most online forums' index pages. This baseline regards a thread with more posts as of higher quality. We also consider the number of users (denoted as #Users) as the second baseline in similar context. The corresponding assumption is that users will tend to gather around high-quality discussion threads more than around low-quality threads. The last baseline we consider is the pure text length of a thread (denoted as Length). Although text lengths of threads do not usually appear in forum index pages, such information can be easily extracted from threads without complex content analysis. Length information has also demonstrated its usefulness in determining the quality of other types of user-generated contents, such as answers[5-6] and product reviews[10]. This last baseline prefers longer threads to shorter threads.

5.2 Thread Ranking Results

We first investigate whether utilizing and combining evidences from different aspects of threads can perform effectively in thread ranking task scenario. Series of thread ranking experiments were carried out by varying the thread quality metric for evaluation. The results are summarized in Table 2. For example, the cell at the intersection of the Proposed row and the ratio column contains the NDCG score of the ranked list based on the proposed method against an ideal ranking using the rating ratio metric. Note that higher NDCG means better performance. For the experiments, we were able to collect 1 936, 1 935, 1 934, and 1 937 valid lists of

Table 2. NDCG Results from Thread Tanking

	Ratio	Abs	Avg	BAvg
#Posts	0.747 9	0.951 7	0.816 0	0.960 2
#Users	0.754 0	0.955 3	0.819 2	0.962 8
Length	0.753 5	0.939 6	0.815 3	0.949 0
Proposed	0.855 6	0.956 6	0.821 4	0.963 9

threads for ranking (i.e., having more than two threads with different scores) when the quality metric was set to rating ratio (Ratio), absolute rating (Abs), average rating (Avg), and Bayesian average rating (BAvg), respectively.

One general observation is that all baseline approaches show convincing results. It is notable that #Posts and #Users, both of which can be obtained directly from forum index pages, show NDCG values comparable to that of Length, which requires additional analysis of threads. All of the three baselines indicate the amount of participation contributed to a thread. On the one hand, these results imply that larger threads with more posts and more users tend to have higher chance of being high quality. On the other hand, they also imply that there are cases in real world where threads in smaller sizes have better quality.

The proposed method, which utilizes various thread features, always achieves higher NDCG value than any of the baselines across all thread quality metrics. The performance improvement of the proposed method over the best baseline on each run has been revealed to be statistically significant using t-test. This result confirms that we can expect performance improvements in thread ranking when quality indicators from various aspects of a thread are utilized. This also means that our method makes better predictions in cases where smaller threads have higher quality than larger ones.

5.3 Pairwise Thread Selection Results

The second set of experiments is designed to investigate whether the proposed method can demonstrate its effectiveness in a more complex pairwise thread selection scenario. Table 3 summarizes the error rate (ER) results. We were able to collect 11 707, 11 299, 10 546, and 12 091 valid pairs of threads for the pairwise selection experiments when the criterion was set to rating ratio (Ratio), absolute rating (Abs), average rating (Avg), and Bayesian average rating (BAvg) respectively. Note that for the results of the pairwise thread selection task, a lower ER value means better performance.

Table 3. Error Rate Results from Pairwise Thread Selection

	Ratio	Abs	Avg	BAvg
#Posts	0.677 0	0.212 0	0.516 2	0.190 4
#Users	0.669 4	0.209 9	0.516 8	0.192 3
Length	0.635 5	0.199 5	0.485 4	0.188 4
Proposed	0.343 7	0.177 4	0.432 7	0.164 2

Regardless of the choice of thread quality metric, the proposed method outperforms all three baselines. These results are analogous to those observed in the previous thread ranking experiments. The difference in effectiveness between the proposed and the best baseline method for each run is also revealed to be statistically significant. This gives us a firm confirmation that the proposed method is effective in discovering high-quality threads.

5.4 Individual Feature Analysis

We now turn to an important question: which of the thread features make most contributions in the prediction of thread quality? Among different feature evaluation schemes, we follow the method presented in [4] and study the importance of individual features by analyzing the learned Ranking SVM functions. More specifically, we analyze the individual feature weights learned from training data. Here we only discuss topmost features with highest absolute learned weights, because the proposed method utilizes more than 50 unique features. Broadly speaking, the topmost features can be considered as influential ones. Moreover, a thread that has large values for features that have relatively high positive or negative weights should be ranked higher or lower respectively in a global ranking of threads. By studying the individual feature weights, it is also possible to expect which types of thread would get a high ranking when the thread quality criterion is set to a particular quality metric.

We first show the feature weights learned in the thread ranking experiment when the criterion of thread quality is set to rating ratio. Table 4 shows the result. For simplicity, we list aliases (which are self-explanatory) of features instead of their full description. The letter in parenthesis in front of an alias refers to its feature type, i.e., S for Surface, T for Time, I for Initiator, and P for Participant.

Table 4. Top Ten Features — Thread Ranking with Rating Ratio

Feature Type	Feature	Weight
(S)	*ThreadAvgWidth*	−7.295 6
(S)	*NumRegisteredParticipant*	5.425 2
(P)	*NumReplyReceived_Participants*	5.189 9
(S)	*AvgPostPerUser*	−4.457 8
(P)	*NumRevisit_Participants*	−4.271 5
(P)	*NumCoParticipant_Participants*	3.833 2
(T)	*TotalReplyTime*	3.652 3
(S)	*NumRevisitedUser*	3.494 7
(P)	*NumReplyWritten_Participants*	−2.840 5
(S)	*NumReplyPost*	−2.385 0

The learned weights are reasonable and make sense intuitively. It appears that the Surface and Participant types are the most influential feature types when we predict thread quality on the rating ratio basis.

Among Surface type features, *ThreadAvgWidth* receives the highest negative weight, which means that a thread in which the shape is spread more horizontally receives large negative ranking score. This seems natural since wide thread widths often indicate that the thread carries a post with a highly controversial content that causes other people to write replies to argue or disagree. *NumRegisteredParticipant* also seems to be important since registered users tend to generate more useful posts than anonymous users. *AvgPostPerUser* receives high negative weight; this implies that if there are two threads with equal number of posts but different number of users, the thread with less users will get more negative weights than the one with more users. It is also interesting that a thread with many reply posts (*NumReplyPost*) receives negative weights, which indicates that the rating ratio metric values quality over quantity. Among Participant type features, it is interesting that the existence of participants that tend to receive more replies from others (*NumReplyReceived_Participants*) and then write replies to others (*NumReplyWritten_Participants*) in a thread is a good sign of thread quality.

We now examine whether the influential features change when the problem is switched to pairwise thread selection. The result is shown in Table 5. New features that are newly promoted to high ranks have △ signs next to their aliases. We observe that many of the Surface and Participant type features which showed strong influence in the previous thread ranking also retain high absolute feature weights. *AvgPostPerUser* receives the highest absolute weight. However, it also appears that in case of pairwise thread selection, the influence of Time type features is greatly emphasized. Such features include the total and average latency time between posts in reply-to relationship (denoted as *TotalReplyTime* and *AvgReplyTime* respectively) as well as the average duration between the initial post and the rest of posts in a thread (*AvgDurationFromRoot*). The promotion of such Time features is as expected, since the threads being analyzed in the pairwise thread selection are the ones with same creation time.

Table 5. Top Ten Features — Pairwise Thread Selection with Rating Ratio

Feature Type	Feature	Weight
(S)	*AvgPostPerUser*	−4.3593
(T)	*TotalReplyTime*	2.6554
(S)	*ThreadAvgWidth*	−2.2796
(P)	*NumCoParticipant_Participants*	2.2213
(S)	*ThreadWidth* (△)	1.7592
(P)	*NumRevisit_Participants*	−1.6283
(T)	*AvgReplyTime* (△)	1.5204
(S)	*NumRegisteredParticipant*	1.4399
(T)	*AvgDurationFromRoot* (△)	−1.3178
(P)	*AvgInitiated_Participants* (△)	1.0928

We now show the feature weights learned in the thread ranking experiment with the thread quality metric set to absolute rating. Table 6 shows the result. The interesting observation we have made here is that the absolute weights of Surface type features have remarkably increased. The most prominent feature is shown to be *NumParticipant*. This result confirms the observation we have made in Table 2 where #Users (equivalent to *NumParticipant*) showed the best NDCG performance among other baselines in the thread ranking experiment across absolute rating metric.

Table 6. Top Ten Features — Thread Ranking with Absolute Rating

Feature Type	Feature	Weight
(S)	*NumParticipant*	15.6452
(S)	*NumRegisteredParticipant*	13.1412
(S)	*ThreadAvgWidth*	5.8022
(S)	*NumReplyPost*	5.6956
(P)	*NumRevisit_Participants*	−4.0014
(P)	*NumInitiated_Participants*	3.9179
(P)	*NumReplyReceived_Participants*	3.4918
(I)	*NumReplyWritten*	−1.8394
(S)	*ThreadWidth*	−1.7972
(I)	*AvgNumParticipantBrought*	1.7801

We also note that *ThreadAvgWidth* and *NumReplyPost*, which previously received negative feature weights on the rating ratio basis, now have positive weights. All of these Surface type features reflect the amount of contributions made toward a thread. Such results imply to us that the absolute rating criterion is biased toward threads with more users and more posts (i.e., values quantity over quality). It is also interesting that not only Participant type features but Initiator type features make appearance in topmost features with highest absolute weights. It appears that a thread initiated by a user who has not participated in other threads for many times before (*NumReplyWritten*) but initiated threads that induced large number of participants in average (*AvgNumParticipantBrought*) has higher chance of being ranked highly on the absolute rating basis.

Table 7 shows the result of feature ranking when the task is switched to the pairwise thread selection in absolute rating basis. Surprisingly, Surface type features almost dominate the topmost rankings. Besides the ones that already received high feature weights in Table 6, features such as *InitPostLength* and *NumPostByAuthorityUsers* are newly appeared. These features with positive weights support the previous observation that absolute rating values quantity over quality. It is also interesting that Time features do not play a crucial role in absolute rating basis.

Table 7. Top Ten Features — Pairwise Thread Selection with Absolute Rating

Feature Type	Feature	Weight
(S)	*ThreadAvgWidth*	5.4178
(S)	*NumRegisteredParticipant*	4.4519
(S)	*NumParticipant*	4.1888
(S)	*ThreadWidth*	−2.6247
(S)	*NumReplyPost*	2.2511
(S)	*InitPostLength* (△)	2.0868
(P)	*NumParticipated_Participants* (△)	1.9459
(S)	*NumPostByAuthorityUsers* (△)	1.9107
(S)	*ThreadDepth* (△)	−1.8693
(P)	*NumCoParticipant_Participants* (△)	1.6367

We now discuss the topmost features for thread ranking based on the average rating metric. The results are shown in Table 8. It appears that Surface and Participant features are more influential than the other types. The feature with the highest absolute weight is revealed to be *NumRegisteredParticipant*. It surprisingly has a high negative weight, which implies that a thread may receive large negative weight if there are too many registered participants in the thread. However, this does not mean that registered users tend to create contents of lower quality than anonymous users; the feature merely represents the absolute number not the ratio. We believe this only indicates that a thread may not be considered of high quality if there are too many users creating posts in the thread, which is reasonable in the case of the average rating since it would not favor large threads with many users as much as the previous absolute rating. It is also noteworthy that a thread with longer initial post will receive higher ranking (*InitPostLength*).

Table 8. Top Ten Features — Thread Ranking with Average Rating

Feature Type	Feature	Weight
(S)	*NumRegisteredParticipant*	−6.6014
(S)	*InitPostLength*	3.7933
(P)	*NumCoParticipant_Participants*	−3.2650
(T)	*AvgDurationFromRoot*	3.1359
(S)	*ThreadAvgWidth*	2.7573
(S)	*AvgPostPerUser*	−2.6059
(S)	*NumRevisitedUser*	−2.4747
(P)	*NumReplyReceived_Participants*	2.1455
(S)	*NumReplyPost*	1.9324
(P)	*NumParticipated_Participants*	1.5797

We observe that a thread will be ranked highly if the participants tend to participate in many threads before (*NumParticipated_Participants*) with small number of users (*NumCoParticipant_Participants*), and receive many replies from others (*NumReplyReceived_Participants*). This intuitively represents the behavior of experts, who contribute to many threads that have not had much contribution yet and also receive many comments from others. Among Time type features, *AvgDurationFromRoot* receives the most weight. This implies that a long-living thread with long durations of time will be predicted of high quality.

Table 9 shows the analysis of topmost features in pairwise thread selection experiment on the average rating basis. Many of the features that appeared previously in Table 8 are replaced by new features in Table 9, but we note that *AvgDurationFromRoot* now receives the highest absolute weight among all other features. This result is also consistent with the observation we made earlier from the pairwise thread selection experiment on rating ratio basis. It is also interesting that *ReplyPostAvgLength* also appears below *InitPostLength*. This implies that the length information of a thread is taken into confidence in the process of thread quality prediction in the average rating criterion. Both length features have positive feature weights, which mean that a thread with longer content length tends to receive higher ranking score than a shorter thread. This result also confirms the previous observation we have made in Table 3 where Length baseline showed the lowest error rate performance among all the three baseline features in pairwise thread selection experiment on average rating basis.

Table 9. Top Ten Features — Pairwise Thread Selection with Average Rating

Feature Type	Feature	Weight
(T)	*AvgDurationFromRoot*	3.0655
(P)	*NumInitiated_Participants* (△)	2.5365
(S)	*AvgPostPerUser*	−2.4623
(S)	*InitPostLength*	2.3847
(S)	*NumPostByAuthorityUsers* (△)	−1.7895
(S)	*NumRevisitedUser*	−1.6304
(P)	*AvgNumRevisit_Participants* (△)	−1.3863
(I)	*AvgNumParticipantBrought* (△)	1.2963
(S)	*ReplyPostAvgLength* (△)	1.1980
(I)	*NumRevisit* (△)	−1.0891

We lastly analyze the feature weights learned when the criterion of thread quality is set to Bayesian average rating. Let us begin with Table 10, which shows the feature ranking results from thread ranking experiment. We find that the result seems to be very similar to that of the thread ranking experiment in absolute rating basis. The top three features with highest weights are revealed to be *NumParticipant*, *NumRegisteredParticipant*, and *NumReplyPost*. This implies that Bayesian average rating also gives high ranking scores to longer threads with many users and many posts. Although

Bayesian average rating of a thread is derived from the average rating of posts, it has the tendency to become closer to the average score of threads if it does not have many posts in it, which means the number of posts still acts as an important factor in the overall rating of the thread. We believe this is the reason that the Bayesian average rating metric shows a tendency more similar to the absolute rating than the average rating.

Table 10. Top Ten Features — Thread Ranking with Bayesian Average Rating Criterion

Feature Type	Feature	Weight
(S)	*NumParticipant*	18.945 3
(S)	*NumRegisteredParticipant*	15.160 7
(S)	*NumReplyPost*	8.308 9
(S)	*ThreadAvgWidth*	7.355 4
(P)	*NumRevisit_Participants*	−4.137 5
(P)	*NumInitiated_Participants*	3.692 9
(P)	*NumReplyReceived_Participants*	3.109 0
(P)	*NumCoParticipant_Participants*	2.087 8
(I)	*AvgNumParticipantBrought*	2.041 4
(S)	*NumPostByAuthorityUsers*	1.998 0

Table 11 shows the analysis of topmost features in pairwise thread selection experiment with Bayesian average rating criterion. Again, the result looks very similar to that of the absolute rating basis. This result supports our previous observation in that Bayesian average rating has a very similar tendency to absolute rating.

Table 11. Top Ten Features — Pairwise Thread Selection with Bayesian Average Rating

Feature Type	Feature	Weight
(S)	*ThreadAvgWidth*	6.904 8
(S)	*NumRegisteredParticipant*	5.307 1
(S)	*NumParticipant*	5.254 0
(S)	*NumReplyPost*	2.811 3
(S)	*ThreadWidth*	−2.754 8
(P)	*NumParticipated_Participants*	2.463 8
(P)	*NumCoParticipant_Participants*	2.128 2
(S)	*NumPostByAuthorityUsers*	2.074 6
(S)	*InitPostLength*	1.980 2
(S)	*ThreadDepth*	−1.870 0

5.5 Feature Type Comparison

Here we report the effectiveness of the different feature types on predicting the quality of threads. Various Ranking SVM models were learned with individual feature types. Table 12 and 13 summarize the results. Bold figures mean the best performance within the same thread quality metric. Here we find that Surface type features outperform the other three types in both task scenarios without being affected by the metric of thread quality. This observation implies that the evidence extracted from the surface of a discussion thread contributes the most to the effective prediction of its overall quality. Another interesting observation is that the model learned only with Initiator type features performs worse than the model learned with Participant type features. This indicates that a thread initiated by an authoritative user does not necessarily lead to a high-quality discussion. This also might be due to data sparsity since typically only a small number of initiators are also active participants.

Table 12. NDCG Results of Individual Feature Types in Thread Ranking Experiments

Feature Type	Ratio	Abs	Avg	BAvg
Surface	**0.854 5**	**0.956 3**	**0.817 1**	**0.963 3**
Time	0.845 1	0.939 3	0.807 9	0.949 0
Initiator	0.801 6	0.860 9	0.799 2	0.877 1
Participant	0.836 7	0.945 5	0.796 4	0.953 2

Table 13. ER Results of Individual Feature Types in Pairwise Thread Selection Experiments

Feature Type	Ratio	Abs	Avg	BAvg
Surface	**0.343 8**	**0.175 1**	**0.444 9**	**0.160 4**
Time	0.361 0	0.218 0	**0.444 9**	0.202 8
Initiator	0.460 0	0.347 1	0.497 2	0.346 9
Participant	0.398 5	0.216 5	0.481 2	0.204 7

6 Related Work

The most related work to ours is the study on predicting the quality of posts in online forums. [12-13] present a binary classifier that predicts whether the quality of a forum post is good or bad by using both linguistic and forum-specific features. [14] extends the research by building a classifier that predicts the rating of forum posts in a finer level (i.e., low, medium, and high). Their work uses a relatively larger dataset for evaluations and reports that non-textual features yield better performance than textual features, which is noteworthy. [15] proposes to measure the quality of forum posts by evaluating the post usage behavior of the forum community. Also, [16] proposes a binary classifier that predicts whether a forum comment is of high quality or not by capturing information about its length and how it is situated within a series of comments. These studies focus on individual post as the primary unit of quality assessment, whereas our work is concerned with finding high-quality threads, which is a series of posts built on a same topic. Combination of the two approaches would be an interesting direction for future work.

There also have been studies on forum content retrieval. [2] evaluates several algorithms for thread retrieval and finds that utilizing thread structure demonstrates better retrieval performance. [3] investigates how the discovery of reply structures in threads can improve the performance of both thread and post search. [1] introduces a thread retrieval model that leverages thread structure using inference networks and proposes few non-textual features that help improve retrieval performance. Our research is related to these approaches in that we utilize thread structure information for assessing thread quality. However, we do not compare our approach with existing forum retrieval algorithms, because there is no query involved in the task scenarios addressed in this paper.

Other recent studies on mining online forums include the recovery of reply-to structures in a forum thread[17], the mining of question-answer pairs in forums[18], the finding of experts in forums[19], the automatic summarization of threaded discussions[20], and the extraction of opinions and influential users in forums[21-22]. [23] introduces a recommender system for online forums, where the goal is to recommend discussion topics to users with consideration to the dynamically evolving nature of the forum.

Studies from the E-learning community[24-25] present tools that can visualize some quality indicators of a discussion thread in a computer-aid learning environment to help decrease the manual workload of human moderators who wish to monitor important threads. Some visual quality indicators overlap with the features proposed in this paper, but their effectiveness have not been empirically evaluated on real world forum data with actual ratings.

7 Conclusions

In summary, the contributions of this paper are three fold. First of all, this is the first work to define and address the problem of finding high-quality threads in online discussion sites. We presented two new tasks involving thread quality and considered them as ranking tasks. Second, we proposed new ways to measure the overall quality of a thread based on the aggregation of its post ratings. This is meaningful in that forum archives with post ratings can be used as an alternative resource to learn useful relationships between threads and their overall quality. Lastly, we not only demonstrated on a real world thread data that utilizing variety of non-content features for quality prediction is effective but provided a careful comparative study of the features that has not been reported previously. We observed that many features for predicting thread quality show different degrees of contributions depending on the thread quality criteria. In terms of feature types, features extracted from the surface of threads were observed to be the most effective feature type for predicting thread quality in general.

For future work, we plan to perform extensive validation and comparison of the thread quality metrics since we clearly do not know which of them is better than the others. In particular, we plan to get user preference judgments of threads using crowdsourcing web services and validate whether these human judgments actually correlate with any metric. If we can identify the thread quality metrics that correlate with human judgments, it would be interesting to derive the global ranking of thread features conditioned on those metrics. We also plan to confirm the results by applying other learning-to-rank algorithms besides Ranking SVM and do a more extensive feature comparison study by using not only learned weights but other evaluation schemes.

References

[1] Bhatia S, Mitra P. Adopting inference networks for online thread retrieval. In *Proc. the 24th AAAI*, July 2010, pp.1300-1305.

[2] Elsas J L, Carbonell J G. It pays to be picky: An evaluation of thread retrieval in online forums. In *Proc. the 32nd SIGIR*, July 2009, pp.714-715.

[3] Seo J, Croft W B, Smith D A. Online community search using thread structure. In *Proc. the 18th CIKM*, Nov. 2009, pp.1907-1910.

[4] Joachims T. Optimizing search engines using clickthrough data. In *Proc. the 8th ACM KDD*, July 2002, pp.133-142.

[5] Agichtein E, Castillo C, Donato D, Gionis A, Mishne G. Finding high-quality content in social media. In *Proc. WSDM*, Feb. 2008, pp.183-194.

[6] Jeon J, Croft W B, Lee J H, Park S. A framework to predict the quality of answers with non-textual features. In *Proc. the 29th SIGIR*, Aug. 2006, pp.228-235.

[7] Gómez V, Kaltenbrunner A, López V. Statistical analysis of the social network and discussion threads in slashdot. In *Proc. the 17th WWW*, April 2008, pp.645-654.

[8] Joachims T. Making large-scale SVM learning practical. In *Advances in Kernel Methods: Support Vector Learning*, Schölkopf B, Burges C J C, Smola A J (eds.), The MIT Press, 1999, pp.169-184.

[9] Järvelin K, Kekäläinen J. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 2002, 20(4): 422-446.

[10] Liu J, Cao Y, Lin C Y, Huang Y, Zhou M. Low-quality product review detection in opinion summarization. In *Proc. EMNLP-CoNLL*, June 2007, pp.334-342.

[11] Xu G, Ma W Y. Building implicit links from content for forum search. In *Proc. the 29th SIGIR*, Aug. 2006, pp.300-307.

[12] Weimer M, Gurevych I. Predicting the perceived quality of web forum posts. In *Proc. RANLP*, Sept. 2007, pp.643-648.

[13] Weimer M, Gurevych I, Mühlhäuser M. Automatically assessing the post quality in online discussions on software. In *Proc. the 45th ACL*, June 2007, pp.125-128.

[14] Wanas N, El-Saban M, Ashour H, Ammar W. Automatic scoring of online discussion posts. In *Proc. the 2nd WICOW*, Oct. 2008, pp.19-26.

[15] Chai K, Hayati P, Potdar V, Wu C, Talevski A. Assessing post usage for measuring the quality of forum posts. In *Proc. the 4th DEST*, April 2010, pp.233-238.

[16] FitzGerald N, Carenini G, Murray G, Joty S. Exploiting conversational features to detect high-quality blog comments. In *Proc. the 24th Canadian Conf. Advances in Artificial Intelligence*, June 2011, pp.122-127.

[17] Lin C, Yang J M, Cai R, Wang X J, Wang W. Simultaneously modeling semantics and structure of threaded discussions: A sparse coding approach and its applications. In *Proc. the 32nd SIGIR*, July 2009, pp.131-138.

[18] Cong G, Wang L, Lin C Y, Song Y I, Sun Y. Finding question-answer pairs from online forums. In *Proc. the 31st SIGIR*, July 2008, pp.467-474.

[19] Zhang J, Ackerman M S, Adamic L. Expertise networks in online communities: Structure and algorithms. In *Proc. the 16th WWW*, May 2007, pp.221-230.

[20] Zhou L, Hovy E. On the summarization of dynamically introduced information: Online discussions and blogs. In *Proc. AAAI Spring Symposium 2006 — Computational Approaches to Analyzing Weblogs*, July 2006, pp.237-242.

[21] Morzy M. On mining and social role discovery in Internet forums. In *Proc. SOCINFO*, June 2009, pp.74-79.

[22] Kaiser C, Bodendorf F. Opinion and relationship mining in online forums. In *Proc. WI-IAT*, Sept. 2009, pp.128-131.

[23] Castro-Herrera C, Cleland-Huang J, Mobasher B. A recommender system for dynamically evolving online forums. In *Proc. the 3rd RecSys*, Oct. 2009, pp.213-216.

[24] Bratitsis T, Dimitracopoulou A. Indicators for measuring quality in asynchronous discussion forae. In *Proc. CELDA*, Dec. 2006.

[25] Simoff S J. Monitoring and evaluation in collaborative learning environments. In *Proc. CSCL*, Dec. 1999.

Jung-Tae Lee received the B.S., M.S., and Ph.D. degrees in computer science from Korea University in 2006, 2008, and 2012 respectively. He is currently a researcher at Naver Corporation. His research interests are information retrieval and its applications.

Min-Chul Yang received the B.S. degree in computer science from Korea University in 2010. He is currently a Ph.D. candidate in Korea University. His research interests are in several areas of social network analysis, including social graph analysis and user recommender systems.

Hae-Chang Rim received the B.S. degree from Korea University in 1979, the M.S. degree in computer science from University of Missouri-Columbia in 1983, and the Ph.D. degree in computer science from University of Texas at Austin in 1990. He is currently a professor in the Division of Computer and Communications Engineering at Korea University. His research interests are in areas of natural language processing, including natural language understanding, machine translation, information retrieval, and question answering.