Answer Quality Prediction in Q/A Social Networks by Leveraging Temporal Features

YUANZHE CAI
and
SHARMA CHAKRAVARTY
Information Technology Laboratory
Department of Computer Science and Engineering
The University of Texas at Arlington, Arlington, Texas 76019

Community Question Answering (or CQA) services (also known as Q/A social networks) have become widespread in the last several years. It is seen as a potential alternative to search as using Q/A services avoids sifting through a large number of (ranked) search results, returned by a typical search engine, to get at the desired information. Currently, *best* answers in CQA services are determined either manually or through a voting process. Many CQA services calculate activity levels for users to approximate the notion of expertise. As large numbers of CQA services are becoming available, it is important and challenging to predict *best* answers (not necessarily answers by an expert) using machine learning techniques. Previous approaches, typically, extract a set of features (primarily textual and non-textual) from the data set and use them in a classification system to determine the *best* answer.

This paper posits that temporal features, different from the ones proposed and used in the literature, are better-suited for Q/A data sets and can be quite effective for predicting the quality of answers. The suitability of temporal features is based on the observation that these services are dynamic in nature in terms of the number of users participating in a given period and how many questions they choose to answer over an interval. We propose and analyze a small set of temporal features, and demonstrate that a few of these features work better than the large number of features used in the literature using the same traditional classification techniques. We also argue that the classification approaches measuring precision and recall are not well-suited as the CQA data is unbalanced, and quality ranking of *all* answers need to be predicted. We propose the use of learning to rank approaches, and show that the features identified in this paper work very well with this approach as well. We use multiple, diverse data sets to establish the utility and effectiveness of features identified for predicting the quality of answers. This approach allows us to qualitatively predict the best answer as well as rank *all* answers. The long-term goal is to build a framework for identifying experts, at different levels of granularity such as global and concept-specific, for CQA services.

Keywords: Question/Answer Services, Answer Quality, Temporal Features, Learning to Rank

1. INTRODUCTION

Community Question Answering (or CQA, also termed Q/A social networks) is gaining momentum in the last several years. It is seen as an alternative to search as it avoids dealing with a large number of answers/results as well as the task of sifting through the results, returned by a typical search engine (although ranked), to get at the desired information. Both general purpose and topic-specific communities are growing in numbers for posting questions and obtaining direct answers in a short period of time (or even in real-time). The corpus of previously answered questions is growing steadily and provides an opportunity for extracting good answers if the same or similar question is asked. *Yahoo! Answers*[1] (Y!A), for example, provides a broad range of topics where as *Stack Overflow*[2] (SO), and *Turbo Tax Live*[3] (TT) are quite focused and domain-specific.

In contrast to the traditional search engines such as Google [Brin and Page 1998], CQA services provide an alternative paradigm for seeking targeted information. These communities allow questioners to post questions and others to provide answers. These communities have become

[1] `http://answers.yahoo.com/`
[2] `http://stackoverflow.com/`
[3] `https://ttlc.intuit.com/`

quite popular in the last several years for a number of reasons. First, because of the targeted response from users with knowledge and/or experience, these answers are likely to be more relevant, useful, and easy to understand for the questioner. Second, the question answering communities also provide a consolidated communication environment where answers to related questions can also be viewed. This environment facilitates multiple answers (likely from different perspectives) and discussion (in the form of comments, threads) that can benefit the questioner (and others as well). In some CQA services, it is also possible for the questioner to interact with the answerer (by email or other means) for clarification and advise. This paradigm, although quite different from the instantaneous search for stored information, is likely to provide the questioner with relevant and useful information, and further act as a substitute for search. Finally, the forum provides an incentive for people to showcase their expertise and in the process gets recognized by the community. For this reason, many CQA services allow the questioner to flag the *best* answer from the set of answers. Some CQA services support a voting mechanism to rank the responses. The notion of an expertise level also exists in some services and is based on several factors such as number of best answers given by a user, votes obtained for answers, etc.

Although Naver[4] was the first community question answering service (started in South Korea in 2002), this phenomenon has grown significantly, and currently a large number of CQA services on a variety of topics exist that use this paradigm. The fact that the CQA has become so prolific in less than a decade is clearly indicative of its popularity and effectiveness as an alternative to search. As the number of CQA services grows, they are also available as archives motivating new approaches for searching and selecting archived answers that best match a question. *In order to do this, it is critical to be able to automatically analyze large volumes of data, and predict & rank the quality of answers with respect to a question whether it is on a focused topic or on a broader discourse.* This is even more important when we have to deal with a large, unbalanced number of answers and growing data sets. For example, in *Yahoo! Answers* community, some questions have more than 100 answers.

Most of the extant work for evaluating/predicting the quality of answers are based on a large number of features extracted from the data set, and the use of traditional classification approaches for predicting the *best* answer. There are some efforts aimed at predicting information seeker satisfaction as well [Liu et al. 2008]. This paper does not address human aspects of this community. Jeon et al. [Jeon et al. 2006] extract 13 *non-textual* features from the *Naver* data set and build a maximum entropy classification model to predict the *best* answer. Along the same lines, Shah and Pomerantz [Shah and Pomerantz 2010] extract 21 features (mainly non-textual) from *Yahoo! Answers* community and use the logistic regression and classification model to predict the *best* answer.

In this paper, we propose a small set of *temporal* features and establish their effectiveness for predicting the quality of answers. Based on our analysis of four diverse data sets (Y!A, SO-C, SO-O, and TT), the answerer's current state seems to be important and has a bearing on answer quality. It is not difficult to discern that these communities are dynamic in nature: number of users vary over time, users are free to choose what questions they want to answer, users gain experience as they answer questions, and the current set of active users is relevant to determine the answer quality. In Section 4, we elaborate on the features, intuition behind their choice, and their extraction. We use both traditional classification and learning to rank approaches to establish the effectiveness of these features. We compare our results with features and classification methods used in the literature.

We also argue that the classification approaches currently used are not well-suited for this problem. First, they only classify into best and non-best answers. Second, the data set is highly unbalanced with the ratio of best answer and non-best answer being very small (less than 0.1 in Y!A data set). This makes it difficult to build a good classification method to predict even the *best* answer. Also, ranking of all answers for a question and its accuracy is equally important

[4] `http://www.Naver.com/`

which is not captured by traditional classification approaches. We propose and use learning to rank approaches to identify not only the *best* answer but also to rank *all* answers.

Finally, features [Jeon et al. 2006; Shah and Pomerantz 2010] available/used from the different CQA data sets are also different. For example, the *Yahoo! Answers* community has well-defined user levels whereas *Stack Overflow* data set does not have this information. Since a learning to rank model can integrate different features into a unified ranking framework to retrieve high quality answers, we propose the use of learning to rank framework for CQA services. Based on the above observations, we argue that the learning to rank approaches are better-suited for this problem. This is further elaborated in Section 6.

The focus of this paper is two fold: (i) to identify and analyze a set of new features that can be used for multiple and diverse (in terms of topics covered and other characteristics) data sets for predicting and ranking answer quality and (ii) to propose and demonstrate the appropriateness and applicability of learning to rank models for predicting best answer as well as ranking all answers in Q/A data sets. We want to clarify that we are not proposing a new learning to rank model but argue for that approach and use one to substantiate our reasoning for that approach. This paper builds upon our earlier work.

Contributions: The contributions of this paper are:

i) Identification and justification of *temporal* features that are relevant to multiple, diverse data sets.
ii) Demonstrate the superiority of *temporal* features for answer quality as compared to the features used in the literature using the same classification approach.
iii) Argue for the learning to rank model as a better approach for measuring the quality of answers and for ranking all answers in CQA data sets.
iv) Extensive experimental analysis of four different and diverse data sets using the proposed features and a learning to rank model.
v) Results confirming that the proposed features as well as the learning to rank model are effective in ranking answer quality.

The Remainder of the paper is organized as follows. Section 2 analyzes related work relevant to our problem and discusses the differences. Section 3 motivates and defines the problem of answer quality analysis. Section 4 introduces the need for temporal features, elaborate on proposed features, and their computation. In Section 5, we compare our features with extant work for inferring answer quality to illustrate significant improvement in accuracy. Section 6 introduces the need for learning to rank approach used along with detailed experimental analysis and discussion of results for four data sets. In Section 7, we summarize the answer quality problem for these Q/A communities and Section 8 has conclusions and future work.

2. RELATED WORK

We categorize previous work related to our problem into three main categories: web page quality analysis, answer quality analysis, and expertise analysis in online communities.

2.1 Web Page Quality Analysis

Features have been used extensively for determining the quality of a web page and our problem is similar, but not exactly the same. In the context of the web, features have been classified by Strong et al. [Strong et al. 1997] into four categories: contextual, intrinsic, representational, and accessibility. Although link analysis [Page et al. 1999; Kleinberg 1999; Cho and Adams 2005] is widely used for ranking web pages, features have also been proposed to determine the quality of web pages by Zhu and Gauch [Zhu and Gauch 2000]. In their approach, documents are first marked manually at different quality levels, such as “good”, “normal” and “bad”. Then, they build a classification model based on these features to predict the quality of other documents.

Clearly, this classification evaluates the web page quality *globally*. However, our problem is slightly different as we need to assess quality of answers to each question.

2.2 Answer Quality Analysis

There is not much work on estimating answer quality in CQA services. Jeon et al. [Jeon et al. 2006] was the first to describe the answer quality problem and propose a maximum entropy approach to predict the quality of answers using non-textual features. They do experiments using the *Naver* online community and demonstrate that it is possible to build a classification model to predict the answer quality. Shah and Pomerantz [Shah and Pomerantz 2010] use a number of automatically extracted features (most are meta-information features) from the *Yahoo! Answers* community to build a classification model. However, both consider only a single data set. Our focus in this paper is on identifying common features for multiple, diverse data sets with differing characteristics. We also show that our features can significantly improve upon earlier results. We also argue for a different approach for answer quality evaluation and establish the efficacy of our features. With a different focus, Harper et al. [Harper et al. 2008] discuss relationship between personal behavior and answer quality, and they conclude that a fee-based system receives better quality answers.

Other related work [Bian et al. 2008; Surdeanu et al. 2008] focus on finding relevant question-answer pairs from Q/A archives for a new query. Both papers integrate user feedback and interactions information (in addition to features) to predict the relevant question-answer pairs. Their focus is to identify similar questions along with their answers. Our research problem is somewhat different in that we are identifying the best answer from the answers given for that query. Our approach also differs in that we are interested in identifying generic features that can be used for diverse data sets. Moreover, none of the related papers propose the use of temporal features in the context of predicting or ranking answer quality.

2.3 Expertise Analysis in the Online Community

There is work on expertise analysis mainly using link-based approaches which is not directly related to this work. Using question-answer network, Zhang et al. [Zhang et al. 2007] and Jurczyk and Agichtein [Jurczyk and Agichtein 2007] build the citation matrix and calculate the authority by PageRank [Page et al. 1999] and HITS [Kleinberg 1999] algorithms, respectively. Their experiments show better results as compared to the application of basic statistical methods. Campbell et al. [Campbell et al. 2003] and Dom et al. [Dom et al. 2003] use link-based ranking algorithms in addition to content-based analysis to rank users' expertise. They apply several link-based algorithms, including PageRank and HITS, to both a synthetic network and a small email network to rank correspondents according to their out-degree of expertise on subjects of interest. In their experiments, link-based algorithms achieve higher accuracy than content-based algorithms.

Our focus in this paper is to automatically rank answer quality for each question so that we can not only infer the best answer but also rank all answers to facilitate retrieval of top-k answers. Our approach is also feature based, but proposes generalization of features to include the temporal aspect as it seems important for these dynamic services. We also rank *all* answers using a learning to rank model and establish its appropriateness for this problem.

3. PROBLEM CONTEXT

Based on current research in this area and our analysis of these communities, we can broadly categorize existing online CQA services based on the interaction by the questioner and the answerer with the service as described below. Of course, the goal of each Q/A service is to provide high quality answers and favourable experience to its users.

3.1 Expert Selection Approach

This approach uses strict guidelines for adding a person as an expert to the Q/A community. After a potential expert joins the Q/A community, s/he has to write a detailed self-introduction

and include credentials. Other users of the community evaluate the expert's self-introduction, background, and questions answered to determine his/her expertise. In this environment, a question has only one answer and because of the strict expert evaluation, these Q/A communities are likely to provide a good/quality answer for a question. Examples of such communities include: *AllExperts*[5], *Google Answers*[6], *MadSci Network*[7]. For the *Allexperts* web site, one is expected to fill an application form which asks for experiences, organizational affiliation, awards received, and publications in relevant areas. After one is chosen as an expert, the community will further evaluate these experts from several aspects, such as knowledgeability, clarity of response, politeness, and response time. Based on this, a questioner can direct his/her questions to one of the chosen experts and receive high quality answers from that expert. Furthermore, in order to retain these experts, these communities provide incentives in the form of bonus, or as in Google Answers, the questioners can quote his/her price for answers.

3.2 Wikipedia Approach

This approach is based on Web 2.0. For each question, the first contributor will answer the question and others are allowed to modify an earlier answer to add their opinion/answer. In this approach, a question has only one answer but is the result of refinement by many answerers. This is in contrast with the traditional approach where a question has many distinct answers (some similar to the others). This approach avoids information redundancy and is beneficial to the questioner as it provides a revised final answer. *Answers*[8] is an example of this approach. In order to confirm the quality of an answer, after the other users revise the answer, *Answers* will permit users to give a trust score to the contributors. Higher trust is placed in the contributor with a better (trust) score.

3.3 User Vote-Based Approach

As the name suggests, this approach evaluates the quality of an answer by the number of votes it receives. This method is widely used in the Q/A communities, but different communities use different strategies. *Yahoo! Answers*, for example, contains three steps to decide the best answer. The first step is the answer collection step. The questioner will post a question and optional description to a specific category. The question then appears in the most recent open questions list in that specific category (and also appears in the questioner's friends' network web page). The question in the open questions list can be answered by the people in the community (and the question in the questioner's friends network web page can be answered by their friends). The second step is the voting step. At this stage, no additional answers are allowed. The answers are listed in random order and other users (than the questioner and answerers) will vote for the best answer. A vote is either a +1 if the voter likes the answer, -1 if the voter does not like the answer. The third step is the best answer selection step. After some fixed period time, the question is closed and the answer with the highest number of votes is chosen as the *best* answer.

In other Q/A communities, such as *Stack Overflow*, *Blurtit*[9], and *Turbo Tax Live*, *Answerbag*[10], there is no clearly-defined time period. A user can answer the question, vote for the answer and choose the best answer as well.

3.4 Questioner Satisfaction Approach

In this approach, only the questioner will decide the answer's quality. If the questioner is satisfied with the answer, s/he can designate it as the *best* answer and send feedback rating which can also include textual feedback. Because the best answer is only decided by the questioner, compared

[5] http://www.allexperts.com/
[6] http://answers.google.com/answers/
[7] http://madsci.org/
[8] http://wiki.answers.com/
[9] http://www.blurtit.com/
[10] http://www.answerbag.com/

with user vote-based approach, the *best* answer resulting from this approach is very subjective. This method is also used in *Yahoo! Answers*.

In summary, in many CQA services, the above-mentioned approaches are *not* mutually exclusive. For example, *Yahoo! Answers* uses both questioner satisfaction approach and user vote-based approach to ascertain the answer quality. *Stack Overflow* allows users to vote the best answer for a question or modify an earlier answer to add their opinions. Many of these communities, such as *Stack Overflow* and *Turbo Tax Live*, also enroll some real experts to periodically post questions and answer questions. In order to develop automated techniques for predicting the *best* answer, and ranking of *all* answers, it is important to understand the above differences and the basis used for determining *best* answers.

3.5 Problem Definition

This work focuses on the user vote-based approach as it seems to be widely used, is consensus-based, and easy to support using the web framework.

Given a question Q_i, and a set of its answers $\{A_{i_1}, A_{i_2}, ..., A_{i_n}\}$, our goal is to calculate the answer quality for each answer using temporal (and other) features and choose the highest ranked one as the *best* answer. Ranking of all answers will allow us to order answers with respect to quality and provide top-k answers for a chosen k. We use the vote-based approach (commonly used by previous work) for comparing our results with the actual votes given for each answer. In case of a tie, all tied answers are *not* considered as *best* answers. The original order of extracted answers is used. *We have purposely chosen this approach to show that even the worst case scenario results in good accuracy.*

It has been established in [Shah and Pomerantz 2010] that manual assessment of quality of answers using a number of subjective criteria is comparable to the vote-based best answer. Hence, in this paper, we test the accuracy against the vote-based approach. Ranking of answers can be used for searching the archives to identify the best (or a few) answer that is consistent with the voted scheme.

4. FEATURE SET AND ITS ANALYSIS

Features are widely used and have been shown to be effective for analyzing documents, images, and web pages, to name a few. So, it is not surprising that this approach has also been used for analyzing questions and answers for predicting not only answer quality but other aspects such as questioner satisfaction as well.

A number of features (mostly non-textual) have been identified in the literature for predicting answer quality. In [Jeon et al. 2006], they show that non-textual features, such as answerer's acceptance ratio, questioner's self evaluation, number of answers, click counts, users' recommendation, and others (there are a total of 13 feature which are extracted primarily from the best answer; some of these features are specific to the *Naver* data set) can be systematically and statistically processed to predict the quality of answers. They assume that the user will randomly generate a "Good" or "Bad" label to each answer. Thus, they build the maximum entropy and kernel density functions to predict these labels. Their experiments conclude that it is possible to build a prediction model to predict "Good" or "Bad" answers for the online Q/A community.

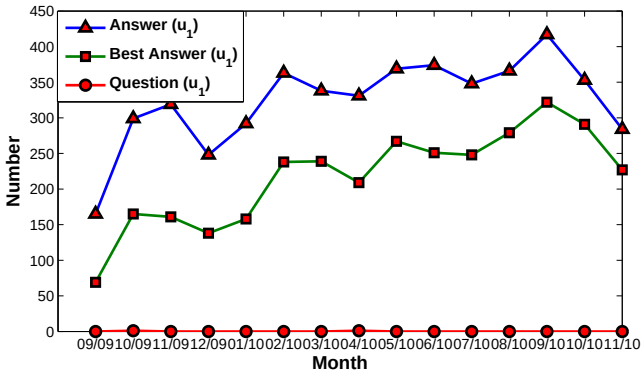
In [Shah and Pomerantz 2010], a number of features (again, most of them being non-textual) are used to train a number of classifiers to predict the best answer. They initially perform a manual assessment of answers using the *Amazon Mechanical Turk*[11] and establish that the qualitative subjective criteria used for establishing the best answer using the *Amazon Mechanical Turk* is comparable to the best answer chosen in the service. Actually, they propose and extract 21 features for each question and answer in the *Yahoo! Quest* data set. Some of the features used are: length of question's subject, information from asker's profile, reciprocal rank of the answer

[11] `https://www.mturk.com/mturk/welcome`

in the time order of answers for the given question, and information from answerer's profile. As this research is the closest to our work, we compare our results with their results in Section 5.

4.1 Need For Temporal Features for CQA Data Sets

Although most of the earlier work (discussed above and in Section 2) focused on non-textual features, our analysis of various CQA services and modality of their usage indicate that there is a strong temporal component that plays an important role and influences answer quality. The number of users, for example, as well as their activity level varies significantly over time.



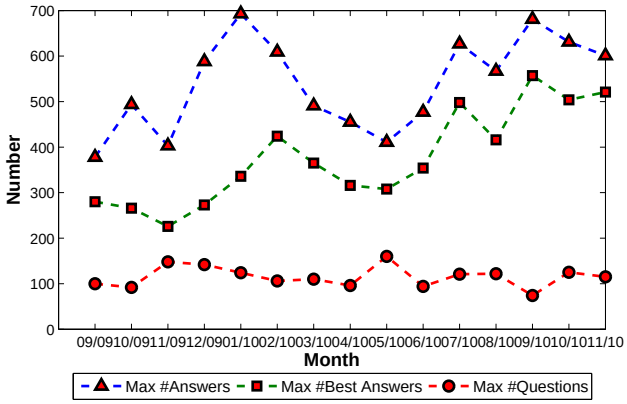
Monthly Single User Activity Level from the SO Data Set

As an illustration of this point, consider Figure 1, which shows a single user activity from the *Stack Overflow* data set. This user, registered in February 2009, is a software engineer with 10 years of experience and is specialized in Java and Agile programming. Figure 1 shows, for each month, the total number of answers, best answers, and questions by that user. The number of answers increased from 168 in September 2009 to 417 by September 2010. The number of best answers also fluctuated in unison with the number of answers given by this user. This user never asked a question in the time period shown. As can be seen, the activity level fluctuates considerably over time and this is to be expected (and is representative of) of a free CQA service where users provide answers at their convenience. Although we have shown a single user, we have observed this phenomena for a significant number of users in the data sets we are using. This seems to be true irrespective of the topic or the focus of the group.

Since it is difficult to show this statistic as an aggregate for all users in a data set, Figure 2 shows the maximum numbers of answers, best answers and questions by any user for each month between September 2009 to November 2010. The fluctuation of these values over this period gives some indication of the widespread nature of user activity changes over a period of time.

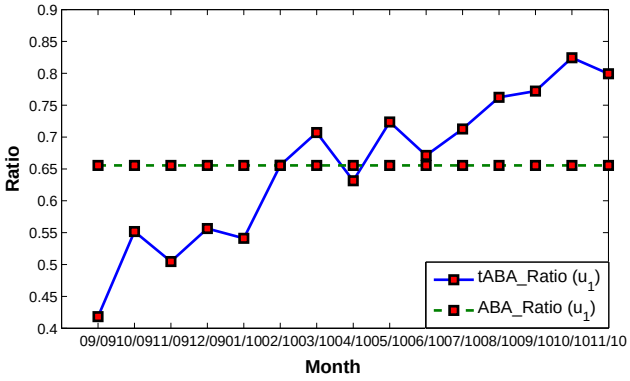
An important aspect of the above observation is how it affects the features extracted for a data set. To illustrate this, Figure 3 shows the feature *best answer ratio* (tABA_Ratio, elaborated below) for the **same** user shown in Figure 1. It is easy to see that the best answer ratio tABA_Score (of an answerer) changes over a period of time (actually increases in this case as the user seems to acquire experience in providing better and easy to understand answers). Contrast this with the same feature value calculated for the entire period (as ABA_Ratio) instead of each month which is also shown as a constant line in Figure 3. Use of this feature value (instead of the temporal one) misses out on the subtle changes to the feature and hence to the accuracy computation.

In fact, temporal features can be viewed as a generalization of its non-temporal counterpart. Instead of computing a feature over the entire data set, it is now computed using smaller relevant intervals to reflect feature values more accurately. This is done for those features that are affected



Aggregate Monthly User Activity Level from the SO Data Set

by the time component. Not all features need to be computed temporally. For example, answer length and similarity score between answer and question reflects temporal component. In addition to activity levels, we have also observed user interest shift (or drift)[12] over time in all the four data sets. This is to be expected as the focus of questions is likely to change over a period of time. Also, what questions a user chooses to answer is also likely to change over a period of time (either based on topic drift or based on user interest drift or both). This feature is likely to be relevant to compute the expertise of a user, but does not affect answer quality as it is already taken into account by the temporal aspect of the features used. Hence, we do not discuss this further in this paper.



Monthly tABA_Ratio and ABA_Ratio for the Same User from the SO Data set with Δt as one Month

Based on the above observations, we believe that answerer's temporal characteristics can significantly contribute to the quality of an answer.

[12]One user in *Stack Overflow* community explains this as follows: *As I used Java language to build my web project in the last quarter of 2009, I am very familiar with that language; but I changed my job and in my new job I develop database applications using the C language. Now, I am interested in C programming questions.*

4.2 Need for Δt and its Computation

Below, we propose a number of features that are a generalization of traditionally used non-temporal features. As the name implies, all the temporal features use a time interval (termed Δt) over which the feature is computed from the data set (in contrast to the non-temporal features that are computed for the entire duration of the data set). we first discuss the intuition behind each feature, its role, and how it is computed from a data set.

Due to the dynamic nature of Q/A services, we need to capture the activity levels of users' around the time when a question is asked and the time period over which that question is answered[13]. Thus, our period of interest starts when a question is asked and ends at the time the last answer is given for that question. This interval is used for determining the quality of an answer given by a user. This is based on the intuition that the quality of answer varies over different periods of time even for the *same* answerer (as we have demonstrated in Figures 1, 2, and 3).

It is also important that this interval (Δt) is chosen properly and is relevant to a data set. Δt needs to capture the current flow of activity in the data set. Thus, for a given data set, we calculate Δt as the interval starting from the time at which the question is asked to the *average time* it takes to receive the last answer in that data set. Note that the average time is specific to a data set.

Table I: Average Response Times for Data sets

Data set	Average time for First Answer	Average time for Best Answer	Average time for Last Answer
Y!A	00:21:15	21:09:27	2 days 12:57:50
SO-C	01:12:33	9 days 20:11:17	12 days 20:32:16
SO-O	01:16:12	9 days 21:57:17	13 days 16:42:18
TT	15:09:17	11 days 19:14:29	26 days 13:10:55

[1] SO-C includes all questions tagged as "C".
[2] SO-O includes all questions tagged as "Oracle".

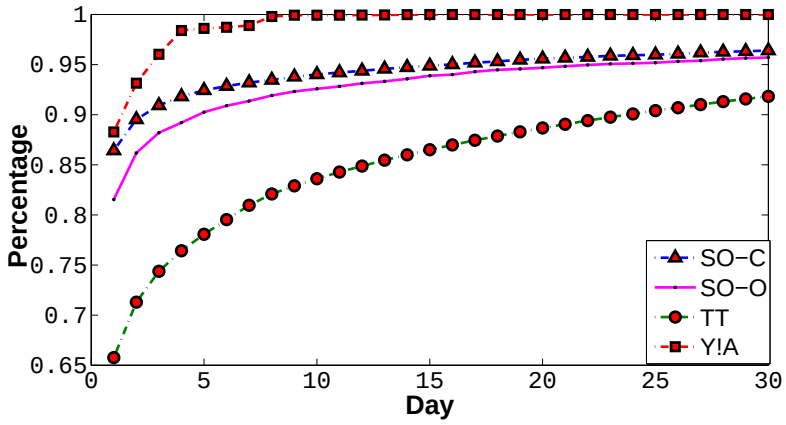
Table I shows the average response times for the four data sets for: the first answer, the last answer, and the best answer. *Yahoo! Answers* receives the last answer in about 2.5 days on the average. In contrast, *Stack Overflow* (e.g., SO-O and SO-C) and *Turbo Tax Live* communities take, respectively, 14 and 26.5 days. Thus, we choose Δt to be 3, 14, and 26.5 days, respectively, for *Yahoo! Answers*, *Stack Overflow*, and *Turbo Tax Live* communities. It is interesting to note that the best answer seems to come in much earlier than the last answer.

The appropriateness of Δt as the average response time for the last answer is elaborated in Figure 4. For these four data sets, Figure 4 plots the aggregate percentage of last answers received over a 30 day period (X-axis) for all questions in the data set. The Y-axis shows the cumulative percentage of last answers for each day. We can see that more than 60% of last answers were received on the first day (65.7% for TT data set, 86.4% for SO-C data set, 81.5% for SO-O data set and 88.2% for Y!A data set). As the time progresses, the last answer percentage for each day decreases significantly. At the average time for the last answer (27 days for TT data set, 14 days for SO data sets and 3 days for Y!A data set), we can see that more than 90% of questions would have received the last answer (91.1% for TT data set, 94.3% for SO-C data set, 93.1% for SO-O data set and 96.2% for Y!A data set).

4.3 Proposed Temporal Features

The following five proposed temporal features are computed for each data set. Below, we describe each feature, its relevance, and how it is computed. Features starting with a t are temporal

[13] It is also possible to use a time period around the question instead of from the question. This is part of our future work.



Cumulative Last Answer Percentage for all Questions (for 30 days)

features computed using the Δt discussed earlier.

1) **Best Answer Ratio (tABA_Ratio) for an Answerer:** For a given answerer u_i and an interval Δt, the tABA_Ratio is the number of best answers to the total number of answers given by that user in that interval. Formally,

$$tABA\_Ratio(u_i, \Delta t) = \begin{cases} 0 & if \quad tAA\_Count(u_i, \Delta t) = 0 \\ \frac{tABA\_Count(u_i, \Delta t)}{tAA\_Count(u_i, \Delta t)} & otherwise \end{cases}$$

where tABA_Count(u_i,Δt) represents the number of best answers by user u_i during Δt and tAA_Count(u_i,Δt) is the number of answers by user u_i during Δt. tABA_Ratio, value $\in [0, 1]$, captures the quality of user's answers. A tABA_Ratio of 1 indicates that each answer is a best answer and a tABA_Ratio of 0 indicates that none of his/her answers are best answers. Fluctuations in tABA_Ratio can also indicate user's effectiveness for quality of answers over a period of time or even interest drift (due to job change, etc.).

2) **Question Answer Score (tAQA_Score) for an Answerer:** This measure classifies each user as: questioner only, answerer only, or a combination thereof. Again, this score can have significant influence over the quality of answers. For example, a user who is an answerer, has a high tABA_Ratio, is likely to provide a better answer. This score is computed as:

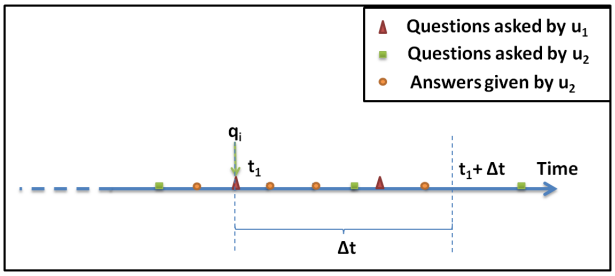
$$tAQA\_Score(u_i, \Delta t) = \begin{cases} 0 & if \quad |A(u_i, \Delta t)| \, and \, |Q(u_i, \Delta t)| = 0 \\ \frac{|A(u_i,\Delta t)| - |Q(u_i,\Delta t)|}{\sqrt{|A(u_i,\Delta t)|^2 + |Q(u_i,\Delta t)|^2}} & otherwise \end{cases}$$

where A and Q indicate, respectively, answers and questions by that user in the specified interval and $|value|$ represents the cardinality of $value$. tAQA_Score describes the level of user participation. A tAQA_Score of -1 indicates a questioner whereas a +1 score indicates an answerer. Along the same lines, we define the rest of the temporal features as follows:

3) **tAA_Count(u_i,Δt)**: Number of answers given by u_i in the interval Δt.
4) **tABA_Count(u_i,Δt)**: Number of best answers given by u_i in the interval Δt.
5) **tAQ_Count(u_i,Δt)**: Number of questions asked by u_i in the interval Δt.

Normalization of the values of some of the above features to the range [0,1] is discussed in Section 6.

Figure 5 illustrates the computation of temporal features using Δt. Figure 5 shows user u_1 asking a question q_i at t_1 and u_2 answers this question in the interval $t_1 + \Delta t$. In order to rank u_2's answer for question q_i, we compute all of the temporal features indicated above for u_2 in the closed interval $[t_1, t_1 + \Delta t]$. As an example, between t_1 and $t_1 + \Delta t$, if user u_2 asks 1 question and answers 3 questions, we calculate tAQA_Score(u_2,Δt) as $\frac{3-1}{\sqrt{1^2+3^2}} = 0.63$. In the same way, we calculate other temporal features for this user u_2.



Computation of Temporal Features

4.4 Other Features

In addition to temporal features, we also extract other features from the data sets. As these overlap with the features from the literature, we do not discuss them but list all the temporal as well as the non-temporal features extracted and their descriptions in Table II. Non-temporal features are computed from the entire data set without using Δt. Temporal features start with t. Features can be understood as question- and questioner-related (start with a Q) and answer- and answerer-related (start with an A).

We extract a total of 22 features, 5 of which are temporal (for an answerer), 5 non-temporal, related to an answerer, 3 non-temporal, related to an answer, 4 non-temporal, related to a question and 5 non-temporal, related to a questioner. Question and questioner features have also been used in the literature [Shah and Pomerantz 2010]. However, our experiments show that they do not contribute to answer quality prediction. We will show, through experiments, in Section 6.2 that the question and questioner features shown in the Table II do not contribute to accuracy.

5. COMPARISON WITH PREVIOUS WORK

We use four diverse data sets (Y!A, SO-O, SO-C, TT) to establish the relevance of 22 features elaborated earlier. We briefly discuss the data sets first to provide their unique characteristics. Table III shows some of the broader characteristics of the data sets used. A subset of these data sets are used for experiments as indicated below.

Y!A Data set: *Yahoo! Answers* community contains 26 top-level topics and a number of sub-topics. The average number of answers for a question is about 10.11 across the entire data set (see Table III). We use "Singles & Dating" category from this data set. The choice of this category is intentional to keep it significantly different from the categories of the other two data sets. In *Yahoo! Answers* community, any user can register as a new user. After becoming a user in the *Yahoo! Answers* community, s/he can ask or answer questions in that community. In addition, once an answer is posted in the community, only the Y!A staff can modify or delete that answer. In the *Yahoo! Answers* community the best answer has already been marked in every resolved question[14] so that we only use resolved questions for the experimental data set.

[14] "Resolved" is one kind of label in *Yahoo! Answers*. If the question has been marked as "Resolved", the best answer has already been identified for this question (also see Section 3 to understand the process for identifying the best answer in the *Yahoo! Answers*).

Table II: Summary of 22 (5 temporal + 17 non-temporal) Features

Feature	Description
Answerer Features (5)	**Proposed Temporal features**
tABA_Ratio	answerer's best answer ratio in Δt
tAQA_Score	answerer's question answer score in Δt
tAA_Count	answerer's number of answers in Δt
tABA_Count	answerer's number of best answers in Δt
tAQ_Count	answerer's number of questions in Δt
Answer Features (3)	**Computed for each Answer (non-temporal)**
A_Length	number of words in the answer
QA_Sim	cosine similarity between question and answer
E_Link	whether or not an embedded link is in the answer
Answerer Features (5)	**Computed for each Answerer (non-temporal)**
AA_Count	number of answers given by an answerer
ABA_Count	number of best answers given by an answerer
AQ_Count	number of questions posted by an answerer
ABA_Ratio	answerer's best answer ratio
AQA_Score	answerer's question answer score
Question Features (4)	**Computed for each question (non-temporal)**
QS_Length	number of words in question's subject
QC_Length	number of words in question's content
Q_Popular	number of answers for this question
Q_Comment	number of comments for this question
Questioner Features (5)	**Computed for each Questioner (non-temporal)**
QA_Count	number of answers answered by questioner
QQ_Count	number of questions posted by questioner
QBA_Count	number of best answers answered by questioner
QQA_Score	questioner's question answer score
QBA_Ratio	questioner's best answer ratio

Table III: Complete Data set Characteristics

Data set	Questions	Answers	Users	Average. # of Answers)
Y!A	1,314,888	13,293,102	1,064,064	10.11
SO-C	25,942	91,615	17,085	3.53
SO-O	8,644	21,879	5,722	2.53
TT	501,978	567,515	486,196	1.13

SO Data set: This service is focused on computer programming topic. Unlike the *Yahoo! Answers* service, *Stack Overflow* community allows a user to modify other user's answers. In another words, when an answerer wants to answer a question, s/he has two choices: modify earlier answer or provide a new one. This community has characteristics of both the Wikipedia approach and the user vote-based approach (discussed in Section 3). As a result, the average number of answers for each question is only 2.36 (See Table III). In our experiments, we only consider the first user who posts the answer as the answerer, because in most cases the first user is likely to provide a larger contribution of the answer than those who revise. *This, again, corresponds to the worst case scenario to illustrate that our approach results in good accuracy.* Each question in this community is marked with a topic tag (e.g., "C" or "Oracle"). We use questions marked as "C" as SO-C data set and questions marked as "Oracle" as SO-O data set.

TT Data set: *Turbo Tax Live* service only discusses tax-related issues. Since tax preparations are made mostly between January and April of each year, this community is very active during these months (that also explains the large average last answer time as our data spans more

than one year). *Turbo Tax Live* community enrolls real tax experts to answer questions. In this community, most of the users are mainly questioners and are less likely to answer questions. Thus, the average number of answers for each question is only about 1.13 (see Table III). Unlike other services, in this service, the same user may give more than one answer to a question. When the answerer gives an answer to a question, the questioner or others are allowed to write comments for this answer and the answerer may give another answer for the same question. This is in contrast to the other two data sets where one answerer can only give one answer to a question. For this data set we choose the answer which has the highest rating as the best answer.

The above data sets serve the purpose of diversity – in terms of topics, mode of interaction, choice of best answer as well as the average number of answers per question. We believe that the diversity of our chosen data sets will stress the features for their effectiveness if they are to perform significantly better than the baseline.

5.1 Comparison with Earlier Work

In this section, we compare the prediction accuracy of answer quality using only temporal features with the features used in [Shah and Pomerantz 2010]. We use the same data set (Y!A data set) and classification approach (a logistic regression model). As described in that paper, we randomly choose 1000 questions in a topic category in which each question has at least 5 answers. For questions with more than 5 answers, we randomly remove non-best answers to bring the number of answers to 5. In *Yahoo! Answers* data set, each question-answer pair has been classified as "Best Answer" or "Non-best Answer". Thus, we build the classification model to evaluate the accuracy of our proposed features. We extract all the 21 features reported in [Shah and Pomerantz 2010] for each question-answer pair and build the **same** logistic regression model using the Weka package[15]. We use 10-fold cross-validation to calculate the accuracy for *Yahoo! Answers* data set.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4010               80.2    %
Incorrectly Classified Instances       990               19.8    %
Kappa statistic                          0.1132
Mean absolute error                      0.2846
Root mean squared error                  0.3782
Relative absolute error                 88.9009 %
Root relative squared error             94.5565 %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.102    0.023      0.526      0.102   0.171      0.721     yes
                 0.977    0.898      0.813      0.977   0.888      0.721     no
Avg.             0.802    0.723      0.756      0.802   0.744      0.721

=== Confusion Matrix ===

   a    b   <-- classified as
 102  898 |   a = yes
  92 3908 |   b = no
```

Accuracy using 21 Features of [Shah and Pomerantz 2010] using Y!A data set

In Figure 6, for 21 features (12 features from Table II by excluding: 5 temporal features and 5 other features (e.g., QQA_Score, QBA_Count, AQA_Score, ABA_Ratio and QA_Sim) not used in [Shah and Pomerantz 2010] plus the following 9 features (e.g., reciprocal rank of the answer in

[15] `http://www.cs.waikato.ac.nz/ml/weka/`

the list of answers for the given question, answerer's star, answerer's point, answerer's level, the number of answerer's solved questions, questioner's point, questioner's star, questioner's level and the number of questioner's solved questions) used in [Shah and Pomerantz 2010], the experiment gives 0.744 classification accuracy (F-Measure score) which is consistent with the results in [Shah and Pomerantz 2010] for 10-fold cross-validation. However, using *only the five temporal* features described in Section 4, the classification accuracy increases to 0.923 as shown in Figure 7 (an improvement of 24%). Our intuition about the importance of temporal features is validated by this comparison between temporal and previously proposed features using the *same* classification method.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4607               92.14  %
Incorrectly Classified Instances       393               7.86  %
Kappa statistic                          0.7645
Mean absolute error                      0.12
Root mean squared error                  0.2478
Relative absolute error                 37.4755 %
Root relative squared error             61.9575 %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.861     0.064      0.772      0.861   0.814      0.958     yes
                 0.937     0.139      0.964      0.937   0.95       0.958     no
Avg.             0.921     0.124      0.926      0.921   0.923      0.958

=== Confusion Matrix ===

   a   b   <-- classified as
 861  139 |   a = yes
 254 3746 |   b = no
```

Accuracy Using Only 5 Temporal Features on Y!A Data set

In order to further understand the incremental effect, we combine the 21 features with the 5 temporal features and measure the accuracy using all 26 features. The accuracy improved only marginally (as compared with the temporal features) to 0.924 as shown in Figure 8. If we interpret this as adding five temporal features to the previously proposed features, accuracy has improved from 0.744 (see Figure 6) to 0.924 (see Figure 8), an improvement of 24.1%. On the other hand, if we are to interpret this as adding 21 features to the proposed five temporal features, there is *no improvement* (from 0.923 to 0.924) in accuracy. This seems to clearly establish the robustness and efficacy of temporal features on answer quality accuracy as compared with previous available results.

Although Shah and Pomerantz [Shah and Pomerantz 2010] use only one data set for their evaluation, we have performed these experiments on the other three data sets. Since SO-C, SO-O and TT data sets do not have four of those features (questioner's level, questioner's star, answerer's level, answerer's star), we only use 17 features in our experiments. The results of these experiments shown in Table VIII for the other three data sets indicate the following: (i) similar to the Y!A data set, F-Measure score for the classification results has improved for the SO data set when the 5 temporal features are added: from 0.843 to 0.896 (6.29%), for the SO-C data set and from 0.839 to 0.892 (6.32%), for SO-O data set. (ii) the F-Measure score for the classification results has improved **much less** for the TT data set from 0.855 to 0.858 (0.35%) when these 5 temporal features are added.

The effect of temporal features depends upon the dynamic nature of the Q/A service. It so happens that the *Turbo Tax Live* service employs a number of experts to answer questions. As

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4610               92.2    %
Incorrectly Classified Instances      390                7.8     %
Kappa statistic                          0.7661
Mean absolute error                      0.1181
Root mean squared error                  0.2472
Relative absolute error                 36.9124 %
Root relative squared error             61.8    %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.861     0.063     0.774      0.861   0.815       0.96      yes
                 0.937     0.139     0.964      0.937   0.951       0.96      no
Avg.             0.922     0.124     0.926      0.922   0.924       0.96

=== Confusion Matrix ===

   a    b   <-- classified as
 861  139 |    a = yes
 251 3749 |    b = no
```

Accuracy Using 5 Temporal + 21 Features on Y!A Data set

the same experts answer questions, user activity levels does not change over a period. We will further discuss the TT data set and the effect of temporal features in Section 6.

Table IV: Accuracy (F-Measure) using 17 Features of Shah and Pomerantz for the other three data sets

Data set	17 Features	5 Temporal Features	17 Features + 5 Temporal Features
SO-C	0.843	0.894	0.896
SO-O	0.839	0.891	0.892
TT	0.855	0.812	0.858

6. LEARNING TO RANK APPROACH FOR ANSWER QUALITY

We argue that the classification method used for comparison of features in Section 5 is not well-suited for the answer quality problem. First, it is difficult to build a good classification model as these data sets are unbalanced. In other words, there is only one best answer for each question, but several non-best answers. For example, in Y!A data set the best answer to non-best answer ratio is about 1 to 9. It is widely recognized in the research community that building a classification model for unbalanced data set is a real challenge. Therefore, it is difficult to find a good traditional classification model for this problem. To illustrate this point, for the classification method used for evaluating features in Section 5, if we categorize *all* the answers as non-best, the accuracy can easily reach 80% (at least 4 out of 5 answers are correctly classified). Hence, the results of [Shah and Pomerantz 2010] which were around 80% were not any better than classifying all answers to be non-best. Hence this binary approach is not meaningful for answer quality ranking.

Second, the best answer is not an absolute best answer in this scenario. In other words, only the answers available for a given question are used for choosing the best answer. Therefore, the best answer choice is relative to other answers. In addition, an answer may be chosen as the best answer due to clarity of expression although it may not be the best answer technically. However, when we build the classification model as described in Section 5, an assumption is made that this question-answer pair is the absolute best answer as compared to all the other pairs. However, this assumption is not generally true for CQA data sets. In contrast, as the learning to rank models build the model for *each* question, this model is likely to be better suited.

Finally, for automated learning of answer quality, it is not enough to identify only the best answer; a ranking of all answers is needed for each question. This is important if one needs to return top-k answers (for a given k) for a question. This becomes even more important for the long-term goal of identifying experts in the system based on the aggregate quality of answers provided by a user (over a period of time). However, the classification method used earlier does not provide answer ranking as it only decides whether or not an answer is the best answer. On the other hand, learning to rank models provide a qualitative value for each answer.

Based on the above observations, we propose the use of learning to rank models to construct answer quality ranking from the question-answer pairs. As the data sets being analyzed can be very large, we choose the *RankSVM* [Joachims 2002] algorithm which has good accuracy in addition to computational efficiency for large data sets. The principle of the RankSVM model is to learn a binary classifier which can discriminate a better answer given a pair of answers for the same question.

We now briefly describe the use of the *RankSVM* algorithm for our problem. Learning to rank approaches require normalized feature values in the range [0, 1]. As some of our features are not in this range (e.g., tAQA_Score), we need to normalize those features that are not in this range. For example, we normalize tAQA_Score which is in the range [-1, +1] by adding 1 and dividing by 2. For other features which are not in the [0, 1] range, we normalize them as follows:

$$NF\_Score(q_i, v_j) = \frac{F\_Score(q_i, v_j)}{max\{F\_Score(q_i, v_j)|j = 1, ..., n\}}$$

where n is the number of values for a feature v_j (e.g., number answers by a user or number of best answers by a user) for question q_i and $NF\_Score(q_i, v_j)$ is the normalized feature score and $F\_Score(q_i, v_j)$ is the extracted feature score for that feature with respect to the question q_i. After normalization, each feature value is between 0 and 1. For example, to calculate the feature tAA_Count for a query q_1 in an interval Δt, we count the number of answers given by different users in that interval. The maximum of those answers form the denominator. The same approach is used for others.

We use pair-wise inputs to RankSVM. For each query q_i, we extract all the answers to form question-answer pairs (we have 5 for each question). For each pair, we extract all the features listed in Table II. We input these extracted features for each question-answer pair along with rank as 1 for the best answer and rank as 0 for non-best answers. Based on this RankSVM derives a model of ranking which maximizes Kendall's τ [Kendall 1938] which is defined as

$$\tau(r^c, r^s) = \frac{P - Q}{\frac{1}{2}n(n-1)}$$

where r^c is the computed rank and r^s is the input rank for the training data set. n is number of elements in rank r^s (which is 5 as we have 5 answers for each question) and $\frac{1}{2}n(n-1)$ describes the number of rank pairs. P is the number of concordant pairs. A concordant pair is one where ranks of pairs from r^c and r^s agree. Similarly, Q is the number of discordant pairs. A discordant pair is one where ranks of pairs from r^c and r^s disagree. Kendall's τ has a value of +1 if the two ranked lists totally agree; Kendall's τ is -1, if the two ranked lists totally disagree; and if two ranks are independent, Kendall's τ score is 0. This is used for maximizing the objective function[16].

Because of the use of the learning to rank approach, we have a ranking of all answers instead of two groups: *best* and *non-best* answers. we can compute the accuracy of any answer with respect to the baseline. This approach will allow us to provide top-k answers to the questioner which is likely to be more useful.

[16]We are using the objective function used by RankSVM without any changes. It is also possible to explore alternative objective functions that are better-suited for this application

6.1 Evaluation Measures

Predicting the *best* answer with good accuracy is important. At the same time, it is equally important to predict the answer quality for all answers and compare that with the voting (or service-specific) approach. Conventional precision and recall does not seem to be appropriate for this approach. Thus, we choose precision at top one (P@1) [Surdeanu et al. 2008; Radev et al. 2002] and the mean reciprocal rank (MRR) methods [Surdeanu et al. 2008; Radev et al. 2002]. For each question q_i, we sort on the predicted ranking values. We pick the top answer and assert this answer as the predicted best answer.

The accuracy of evaluation measures, MRR and P@1, are also related to the number of answers available for each question. With the increase in the number of answers, the accuracy of any ranking model is likely to drop. It is easy to understand that with increase in the number of answers for each question, it becomes more and more difficult to predict the best answer. If every question has only one answer, the worst accuracy of any ranking model will reach 100%; however, if every question has more than 1 answer, the accuracy of ranking of answers is likely to be lower than 100%. Therefore, in our experiments if we choose 1000 questions without considering the number of answers for each question, it is difficult for the other researchers to repeat these experiments because the distribution of number of answers for each experiment is different if we randomly choose these experimental questions. Hence, in our experiments, we **first** randomly choose 1000 questions from each data set which has at least 5 answers for each question. For each question, we retain the best answer and 4 other randomly selected non-best answers. Thus, we use 1000 questions, 1000 best answers and 4000 non-best answers for our experiments. Clearly, in these data sets, if we randomly choose an answer as the best answer (baseline 0), P@1 will be 0.2 and MRR will be 0.46. Any meaningful ranking model should provide better results. *We have also performed experiments by relaxing the five answer constraint (see Section 6.8).*

We use 10-fold cross-validation to calculate P@1 and MRR for each data set and every experiment has been run five times and the average value is reported. Note that we can also compute P@n as we are using a learning to rank model as opposed to the traditional classification approach.

For the accuracy analysis, we use three baselines as we do not have a well-defined baseline for this problem in the literature.

- Random Baseline (**Baseline 0**): We randomly choose an answer as the best answer. This baseline indicates the worst accuracy to predict the best answer and we believe that any approaches should be better than this approach.
- QA_Sim (**Baseline 1**): We use the cosine similarity between a question and its answer to rank the answer. This baseline is widely used in the traditional information retrieval [Salton et al. 1975] and also used in [Littlepage and Mueller 1997] to search for related answers in CQA services. This value can vary depending on the topic and service. We have observed higher similarity value for the TT data set as this service is very focused and the answers are likely to contain common words from the question.
- Shah and Pomerantz's 21 Features (**Baseline 2**): In order to show the effectiveness of temporal features, we also use the 21 feature set used in [Shah and Pomerantz 2010] as one of the baselines. In Y!A data set, we use 21 features as in [Shah and Pomerantz 2010] to identify the answer quality (see Section 5.1). Since SO data set and TT data set do not have four of those features (questioner's level, questioner's star, answerer's level, answerer's stars), we only use 17 features in our experiments. This baseline has been included to show that temporal features will provide significant improvement over the traditional features irrespective of the approach used (*RankSVM* in this case). As many of the features listed in Table II are part of these 21 features, we compare baseline 2 only with the proposed 5 temporal features.

6.2 Irrelevance of Question and Questioner Features

In the literature, [Shah and Pomerantz 2010] use question and questioner features for predicting answer quality and so we have included these features for CQA data sets in Table II. Question and

questioner features are, respectively, the features related to the question and the questioner. Our intuition indicates that these should not have any bearing on answer quality as answer quality is mainly based on answer content, relevance, and the quality of the answerer. Table V tabulates the accuracy of features for the same data sets with and without 9 question and questioner features. We can clearly observe that the *accuracy does not change at all*. This clearly validates our intuition that question and questioner features do not contribute at all to the accuracy; hence, we do not use these 9 features in the rest of experiments discussed below.

Table V: Accuracy With and Without Question/Questioner (Q/Qr) Features

Data set	With Q/Qr Features		Without Q/Qr Features	
	P@1	MRR	P@1	MRR
Y!A	0.810	0.877	0.810	0.877
SO-C	0.535	0.611	0.535	0.611
SO-O	0.536	0.612	0.536	0.612
TT	0.484	0.672	0.484	0.672

6.3 Ranking of Features

In the first set of experiments, we calculate the difference between the average score for the *best* answer and non-best answers and rank each feature by the deviation. Clearly, if the deviation is large, that feature is likely to discriminate better between the best-answer and the non-best answer. Table VI shows the details for the Y!A data set. We show the feature rank for the other three data sets in Table VII. The results of this experiment indicate: (i) features tABA_Ratio and tABA_Count rank quite high in Table VI indicating their discriminating power for answer quality prediction; in fact, tABA_Ratio ranks as number 1 for all data sets (also see Table VII), (ii) feature AQ_Count and tAQ_Count are ranked at the bottom for all data sets and hence does not seem to be very useful for predicting answer quality. This seems logical as users who ask a lot of questions do not seem to contribute to the quality of answers, (iii) as can be seen from Tables VI and VII, the ranking of features is **different** for each data set. This can be attributed to the different characteristics (topic, and others) of the data set, (iv) A_length comes out as an important feature for deciding the answer quality (e.g., top 3 in Y!A data set, top 6 in SO-C data set, top 4 in SO-O data set and top 4 in TT data set). This also indicates that a good answer is likely to be longer as the answerer can explain clearly, and finally (v) E_Link, proposed in [Harper et al. 2008], does not seem to be a good feature for discriminating the best answer from non-best answers. This feature seems to be applicable for Wikipedia-style service than the ones used in this work. In summary, many of the temporal features are ranked high across data

Table VI: Feature Analysis of Y!A Data set

Feature	Mean			
	Best Answer	Non-Best Answer	Difference	Rank
tABA_Ratio	**0.8849**	**0.0681**	**0.8168**	**1**
tABA_Count	**0.7561**	**0.156**	**0.6001**	**2**
A_Length	0.5037	0.2628	0.2409	3
AQA_Score	0.2723	0.1191	0.1532	4
ABA_Ratio	0.2029	0.0807	0.1222	5
ABA_Count	0.1586	0.0812	0.0774	6
QA_Sim	0.349	0.2772	0.0718	7
AA_Count	0.1509	0.0854	0.0655	8
tAA_Count	**0.1625**	**0.1006**	**0.0619**	**9**
tAQA_Score	**0.2016**	**0.1451**	**0.0565**	**10**
E_Link	0.0547	0.0177	0.037	11
tAQ_Count	**0.0972**	**0.078**	**0.019**	**12**
AQ_Count	0.1435	0.1243	0.0192	13

sets. Some non-temporal features (A_Length, ABA_Ratio, for example) also rank high in some data sets.

Table VII: Feature Analysis of SO-C, SO-O and TT Data sets

Data set	SO-C		SO-O		TT	
Rank	Feature	Diff.	Feature	Diff.	Feature	Diff.
1	**tABA_Ratio**	**0.3768**	**tABA_Ratio**	**0.3329**	**tABA_Ratio**	**0.3192**
2	**tABA_Count**	**0.2013**	**tABA_Count**	**0.181**	ABA_Count	0.1588
3	ABA_Ratio	0.154	ABA_Ratio	0.1481	ABA_Ratio	0.1486
4	**tAQA_Score**	**0.0802**	A_Length	0.0673	A_Length	0.1485
5	QA_Sim	0.0661	**tAQA_Score**	**0.0524**	AQ_Count	0.1428
6	A_Length	0.0519	QA_Sim	0.0512	**tAQA_Score**	**0.1069**
7	**tAA_Count**	**0.0455**	**tABA_Count**	**0.1041**	**tABA_Count**	**0.1041**
8	AQA_Score	0.0401	AQA_Score	0.0332	**tAA_Count**	**0.0321**
9	ABA_Count	0.0321	ABA_Count	0.0291	AA_Count	0.0934
10	AA_Count	0.0128	E_Link	0.0111	QA_Sim	0.0284
11	E_Link	-0.005	AA_Count	0.002	E_Link	0.013
12	**tAQ_Count**	**-0.012**	AQ_Count	**-0.0317**	**tAQ_Count**	**-0.0358**
13	AQ_Count	-0.043	**tAQ_Count**	**-0.0525**	AQA_Score	-0.0645

6.4 Ranking of Answers Using Temporal and Other Features

Next, we use the RankSVM learning to rank model for determining answer quality as explained earlier. The random approach (**Baseline 0**) is shown in the first row of Table VIII. We discuss the results with **Baseline 1** (second row of Table VIII) as it is better. It can retrieve less than 25% in top one rank of the correct answer and MRR shows that the correct answer is less than 49%. This baseline only reflects whether this answer is related to this question, but it is difficult to distinguish the answer's quality; hence the accuracy of baseline 1 is somewhat similar to baseline 0. We compare our temporal and other features with these two baselines in Table VIII first. Later, we also present a comparison with a different baseline which uses the features used in [Shah and Pomerantz 2010]. This is elaborated in Section 6.6.

As each feature is added one at a time in *rank order* (note that the rank order is data set dependent), one can observe consistent improvement in accuracy for all data sets. The process is initialized with the QA_Sim (baseline 1) and the learning to rank model incrementally adds

Table VIII: Accuracy Values Compared with Baselines 0 and 1

Features	Y!A		SO-C		SO-O		TT	
	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
Baseline 0	**0.2**	**0.456**	**0.2**	**0.456**	**0.2**	**0.456**	**0.2**	**0.456**
Baseline 1	**0.262**	**0.490**	**0.29**	**0.431**	**0.272**	**0.432**	**0.407**	**0.600**
"+"1	0.791	0.864	0.503	0.587	0.493	0.581	0.407	0.602
"+"2	0.798	0.868	0.505	0.591	0.502	0.585	0.402	0.596
"+"3	0.805	0.871	0.503	0.592	0.506	0.591	0.401	0.595
"+"4	0.811	0.877	0.529	0.607	0.526	0.611	0.478	0.664
"+"5	0.809	0.875	0.531	0.609	0.526	0.611	0.487	0.676
"+"6	0.809	0.876	0.526	0.607	0.526	0.611	0.478	0.669
"+"7	0.809	0.876	0.531	0.613	0.527	0.613	0.491	0.674
"+"8	0.805	0.874	0.529	0.610	0.529	0.614	0.488	0.674
"+"9	0.805	0.874	0.534	0.618	0.533	0.615	0.478	0.669
"+"10	0.809	0.876	0.534	0.617	0.535	0.616	0.477	0.667
"+"11	0.807	0.875	0.534	0.613	0.534	0.616	0.478	0.669
All features	**0.810**	**0.877**	**0.535**	**0.611**	**0.536**	**0.612**	**0.482**	**0.670**

[1] We add one feature at a time in their rank order listed in Tables VI and VII to the baseline 1 for each data set. Because *QA_Sim* is our baseline 1, we just need to add the other 12 features.

the features and computes the P@1 and MRR values. In Table VIII, for SO-C, SO-O and Y!A data sets we can observe that if we only consider four features (e.g., tABA_Ratio, tABA_Count, ABA_Ratio, A_Length), learning to rank model will achieve significant accuracy improvement and the other features seem only to contribute very little to the accuracy. The results of this experiment shown in Table VIII for all data sets indicate the following: (i) the accuracy for the best answer has improved **significantly**, from 0.262 to 0.810 (209% improvement) for the Y!A data set, from 0.29 to 0.535 (84% improvement) for SO-C data set, from 0.272 to 0.536 (97% improvement), for SO-O data set, and reasonably, from 0.407 to 0.482 (18.4%) for the TT data set, (ii) although the baselines are similar for Y!A, SO-C, and SO-O data sets, the improvement in accuracy is still significant when all the features are included, and finally (iii) the baseline 1 (QA_Sim) is higher for the TT data set and results in lesser improvement in accuracy (18.4%). We believe that the lesser improvement (as compared to Y!A) for the SO Data set (both SO-C and SO-O) is due to the data set characteristics of answers containing program code in SO-O and SO-C. In contrast, in the TT data set answers are given by experts and as the same set of people answer all questions, temporal features have a small effect. Also, there seems to be a higher correlation between questions and answers in this data set as it is a narrow domain and word overlap between a question and its answers seem to be more than other data sets.

We also wanted to make sure that the rank order of features are indeed correct, we have tried to add these features in the "reverse" rank order to see their effect. Table IX shows the results of adding each feature at a time in ***reverse*** *rank order* for each data set. It is easy to observe accuracy improvement for each data set as the features are added in reverse rank order. They clearly indicate that: (i) the final accuracy is the same as that of the previous case where features were added in rank order, (ii) for SO-C, SO-O and Y!A data sets, the last bottom three features (AQ_Count, tAQ_Count, E_Link) seems to improve the accuracy *only marginally*, from 0.262 to 0.266 (1.8%) for Y!A data set, from 0.29 to 0.294 (1.4%) for SO-C data set, from 0.272 to 0.274 (0.7%) for SO-O data set and from 0.407 to 0.407 (0%) for TT data set, (iii) when we add the top four features (e.g., tABA_Ratio, tABA_Count, ABA_Ratio, A_Length), the accuracy improves suddenly and significantly [17], and finally (iv) as observed earlier, for the TT data set, proposed five temporal features do not seem to improve accuracy; however, A_Length and QA_Sim (baseline 1) seem to be two good features for this data set.

Table IX: Accuracy Values Compared with Baselines 0 and 1 in **Reverse** Order

Features	Y!A		SO-C		SO-O		TT	
	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
Baseline 0	**0.2**	**0.456**	**0.2**	**0.456**	**0.2**	**0.456**	**0.2**	**0.456**
Baseline 1	**0.262**	**0.490**	**0.29**	**0.431**	**0.272**	**0.432**	**0.407**	**0.600**
"+"12	0.262	0.491	0.292	0.431	0.272	0.434	0.407	0.602
"+"11	0.263	0.493	0.293	0.431	0.273	0.436	0.405	0.598
"+"10	0.267	0.492	0.292	0.432	0.272	0.432	0.407	0.603
"+"9	0.266	0.493	0.294	0.431	0.274	0.438	0.421	0.632
"+"8	0.321	0.507	0.35	0.465	0.301	0.461	0.421	0.631
"+"7	0.421	0.567	0.361	0.481	0.303	0.463	0.422	0.632
"+"6	0.442	0.58	0.362	0.481	0.342	0.482	0.421	0.632
"+"5	0.471	0.6	0.401	0.51	0.351	0.486	0.424	0.633
"+"4	0.513	0.621	0.398	0.508	0.352	0.513	0.478	0.669
"+"3	0.558	0.643	0.405	0.512	0.359	0.519	0.477	0.668
"+"2	0.807	0.868	0.532	0.613	0.531	0.612	0.48	0.671
All features	**0.810**	**0.877**	**0.535**	**0.611**	**0.536**	**0.612**	**0.482**	**0.670**

[1] We add one feature at a time in their **reverse** rank order listed in Tables VI and VII to baseline 1 for each data set.

[17] For example, when we add the tABA_Count for the Y!A data set, the accuracy increases by 44.6% and similarly for others.

6.5 Further Validation of Feature Ranking

In previous experiments, we use the ranking of features shown in Table VII and add them one at a time to observe their effect on accuracy. We also want to compare the top-k features with rest of the features to establish the appropriateness of feature ranking. For the next set of experiments, we classify these 13 features in Table VII into two groups to evaluate the quality of these features. Group 1 includes four top-ranked features (e.g., tABA_Ratio, tABA_Count, ABA_Ratio, A_Length) and group 2 includes the other 9 features (e.g., AQA_Score, ABA_Count, QA_Sim, AA_Count, tAA_Count, tAQA_Score, E_Link, tAQ_Count, AQ_Count). The results of this set of experiments are shown in Table X for all 4 data sets. The results validate the relevance of top ranked features. Table X clearly indicates that if we only consider the four features in group 1, our prediction will achieve the highest accuracy (0.809 for the Y!A data set, 0.534 for the SO-C data set, 0.535 for the SO-O data set and 0.667 for the TT data set) and the other features in group 2 seem very close to the baseline 1 (see Table VIII) without much improvement (0.275 for the Y!A data set, 0.31 for the SO-C data set, 0.292 for the SO-O data set and 0.420 for the TT data set).

Table X: Accuracy Values to Compare with Group 1 and 2

Features	Y!A		SO-C		SO-O		TT	
	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
Group 1	0.809	0.876	0.534	0.617	0.535	0.616	0.477	0.667
Group 2	**0.275**	**0.501**	**0.31**	**0.445**	**0.292**	**0.340**	**0.420**	**0.620**

6.6 Accuracy Comparison Using Previous Work as Baseline

As baselines 0 and 1 were basic baselines that did not really consider answer quality in any way, we wanted to compare accuracy improvement of temporal features with respect to a baseline that took quality into consideration. Baseline 2 (described earlier) uses features from [Shah and Pomerantz 2010]. Baseline 2, shown in the first row of Table XI, shows substantial improvement, as expected, over baselines 1 and 0 as it takes answer quality into consideration in terms of features. It indicates that the best answer can be found more than 40% of the times in top one rank and MRR shows that the correct answer is in the first two answers. Then, as each temporal feature is added one at a time, one can observe consistent improvement in accuracy for all data sets (see Table XI). The best answer accuracy has increased significantly, from 0.552 to 0.813 (47%) for the Y!A data set, from 0.401 to 0.539 (34%) for the SO-C data set, from 0.395 to 0.537 (36%) for the SO-O data set. This seems to clearly establish the robustness and efficacy of temporal features on answer quality accuracy for these three data sets. However, these temporal features improve the best answer accuracy marginally (from 0.461 to 0.482, 4.5%) for the TT data set.

To better understand the characteristics of the TT data set, we computed the same five features both temporally and non-temporally. For temporal value of features, we use Δt as described earlier and compute these features for each answerer using Δt. The non-temporal computation for the same features uses the duration of the entire data set (which is about 3 years) as Δt. This is similar to the comparison of feature values shown in Figure 3 in Section 4.1. The results of these experiments are shown in Table XII. It is clear that the P@1 and MRR values for both temporal and non-temporal counterparts are very close to each other (0.291 and 0.511 for 5 temporal features and 0.311 and 0.529 for 5 non-temporal features). As explained earlier, since in *Turbo Tax Live* experts *employed* by the organization answer questions and hence the community is not dynamic as compared to other communities. This is important to understand and derive from the data set so that feature selection can be done appropriately.

Table XI: Accuracy Value Comparison with Baseline 2

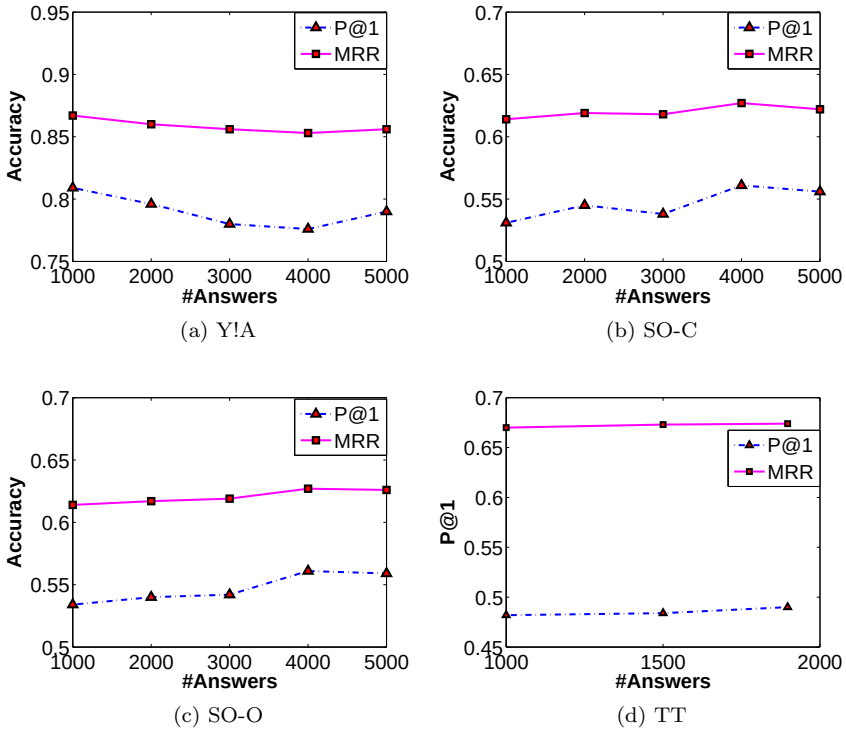
Features	Y!A		SO-C		SO-O		TT	
	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
Baseline 2	**0.552**	**0.640**	**0.401**	**0.510**	**0.395**	**0.511**	**0.461**	**0.659**
"+"1	0.805	0.872	0.532	0.606	0.533	0.604	0.463	0.663
"+"2	0.811	0.874	0.535	0.591	0.532	0.607	0.473	0.667
"+"3	0.811	0.875	0.534	0.592	0.536	0.613	0.479	0.670
"+"4	0.813	0.878	0.537	0.612	0.538	0.612	0.481	0.671
All features	**0.812**	**0.878**	**0.539**	**0.612**	**0.537**	**0.612**	**0.482**	**0.671**

[1] We add 5 temporal feature one at a time in their rank order listed in Tables VI and VII to the baseline for each data set.

Table XII: Comparison of Temporal and Non-Temporal Features for the TT Data set

Features	P@1	MRR
5 Temporal Features	0.291	0.511
5 Non-Temporal Features	0.311	0.529

[1] We use average last answer response time as Δt for 5 temporal features; the same features are computed non-temporally with Δt as the time when the first a question is asked in this community to the time when this community receives the last question (about 3 years for the TT data set used for experiments).



Accuracy Analysis for Larger Data Set Sizes

6.7 Effect on Accuracy with Larger Data Sets

We have performed experiments on data sets whose size (in terms of number of questions and answers) is increasing to make sure our approach preserves accuracy when the data set size is increased. We extract, respectively, 1000, 2000, 3000, 4000, 5000 questions with 5 answer constraint from Y!A, SO-C and SO-O data sets and 1000, 1500, 1897 questions from the TT data set[18]. We only test four features (e.g., tABA_Ratio, tABA_Count, ABA_Ratio, and A_Length) as they have come put to be the top ones in our earlier experiments for predicting answer quality. These experimental results are shown in Figures 9a, 9b, 9b and 9d. They clearly indicate that using larger data sets does not reduce the accuracy to predict the best answer. In Figures 9a, 9b, 9c and 9d the accuracy of these data sets increases slightly with an increase in the number of questions. In Figure 9a the accuracy of Y!A data set drops sightly and recovers with an increase in the number of questions. The reason can be explained as in Y!A data set more data are used as the training data so that this RankSVM model is sightly over-fitting [Everitt and Skrondal 2006] with this data set. Therefore, using 1000 questions as training data seems to be good enough for real CQA data sets.

6.8 Relaxing the Five Answer Constraint

We also wanted to make sure that our features work for real data sets where one cannot assume exactly five answers for each question. This is important but has not been addressed in the literature. If these approaches are to be used on real-world data sets, we need to use all answers for each question (however unbalanced they are).

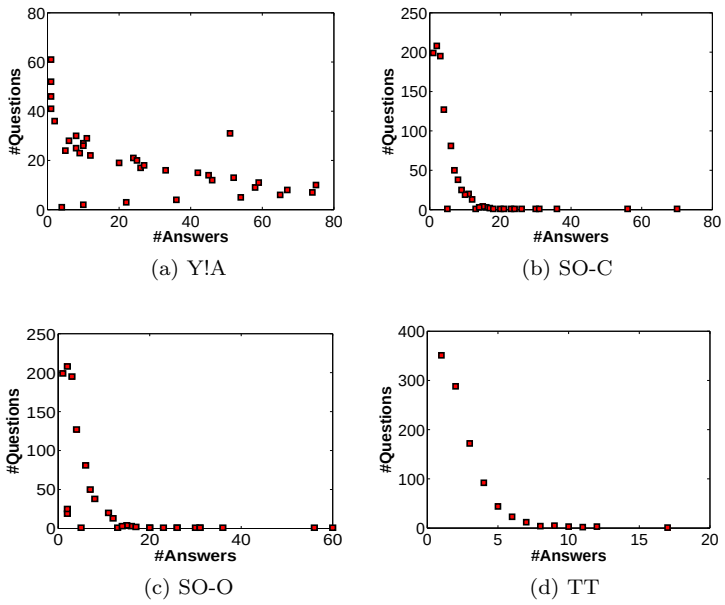
Hence, we performed experiments by relaxing the five answer constraint. We randomly choose 1000 questions from each data set without any constraint on the number of answers. The distribution of answers for these four experimental data sets is shown in Figures 10a, 10b, 10c and 10d. These Figure are very useful for understanding the characteristics of the data sets. For example, it is clear that the TT data set has a larger number of questions with fewer answers as compared to the other data sets.

The results of these experiments, shown in Table XIII, indicate that compared with Table VIII, the accuracy of these data sets without any constraint on the number of answers is even better than the data sets with the 5 answer constraint. Especially for the TT data set, P@1 score increase from 0.482 (see Table VIII) to 0.961 (see Table XIII) and MRR score increases from 0.670 (see Table VIII) to 0.983 (see Table XIII). This is related to the average number of answers for each question which was the lowest at 1.13 for the TT data set (see Table III). So, prediction of accuracy is much better (and easier) as compared to a data set with at least 5 answers. For example, for the TT data set, almost 40% questions have only one answer (see Figure 10d). In the absence of a constrained data set (e.g., with 5 answers), it is difficult to evaluate feature sets.

6.9 Summary

Our experimental results provide confirmation of: (i) learning to rank approach has flexibility to compute accuracy for the desired combination (e.g., best answer, MRR, or others), (ii) temporal features tABA_Ratio and tABA_Count come out to be two most important features that can discriminate the best answer and answer quality even in large and noisy online Q/A data sets, (iii) there is no need to include a large number of features if we choose the feature set judiciously. From our experiments, tABA_Ratio, tABA_Count, ABA_Ratio, and A_Length come out as discriminating features whereas some features, such as tAQ_Count, AQ_Count, and E_Link, do not seem to be useful or make much difference, and finally (iv) since from our experiments for the real CQA data sets using more training questions to build the learning to rank model does not reduce accuracy, we just need 1000 questions as the training data set for real CQA data sets. This is beneficial from a computational viewpoint.

[18]In the TT data set, there are only 1897 questions which have more than or equal to 5 answers



#Questions vs. #Answers in Four Data sets

Table XIII: Accuracy Comparison for 4 Data Sets without Constraint on Number of Answers

Features	Y!A		SO-C		SO-O		TT	
	Top@1	MRR	Top@1	MRR	Top@1	MRR	Top@1	MRR
Baseline 1 (QA_Sim)	**0.273**	**0.501**	**0.428**	**0.562**	**0.447**	**0.581**	**0.957**	**0.980**
"+"1	0.809	0.939	0.626	0.681	0.646	0.715	0.961	0.982
"+"2	0.816	0.939	0.629	0.707	0.641	0.723	0.961	0.983
"+"3	0.816	0.944	0.632	0.724	0.647	0.742	0.961	0.983
"+"4	0.827	0.948	0.664	0.729	0.681	0.751	0.961	0.983
"+"5	0.827	0.946	0.658	0.729	0.675	0.747	0.961	0.983
"+"6	0.826	0.947	0.657	0.721	0.677	0.749	0.961	0.983
"+"7	0.825	0.947	0.651	0.728	0.682	0.751	0.961	0.983
"+"8	0.82	0.944	0.649	0.731	0.667	0.748	0.961	0.983
"+"9	0.821	0.945	0.653	0.735	0.671	0.752	0.961	0.983
"+"10	0.822	0.944	0.653	0.733	0.672	0.751	0.961	0.983
"+"11	0.823	0.944	0.654	0.733	0.674	0.752	0.961	0.983
"+"12	**0.825**	**0.949**	**0.662**	**0.734**	**0.681**	**0.761**	**0.961**	**0.983**

[1] We add one feature at a time in their rank order which is listed in Table VI and VII to the base line for each data set.

7. DISCUSSION

In this paper, we identify an important characteristic, "user behavior", in the Q/A social community. Users take part in social networks voluntarily and communicate with each other. Therefore, for most problems that come under the purview of social community, it is important to consider user behavior as an active ingredient. Q/A community is a dynamic social community where users post questions and other users answer these questions. Because most of Q/A communities are open communities, any user can post his/her questions or answers in these communities. As with any dynamic community, current status of a user plays an important role and has a bearing on answer quality.

We have captured this user behavior as a set of temporal features which help us incorporate

user activity, which can change over a period of time, into a feature value. This is in contrast to the traditional features that are calculated over the entire duration of the data set. For traditional applications that use features, user behavior may not be relevant or may not change. But for applications such as CQA services it makes a significant difference as we have been able to demonstrate convincingly in this paper.

Temporal features can also be viewed as a generalization of traditional non-temporal features. They reflect subtle aspects of the data set for a given granularity. Typically, these communities are used on a as needed basis by users. Answerers do this voluntarily as a service based on their current interest and availability of time. Hence, depending on the service this aspect may be significant or not so significant. However, temporal features are useful and capture user activity and behavior. Some Q/A communities, such as *Turbo Tax Live* community (see Table XI), employ experts to answer questions. Because these communities use experts, these experts answer questions regularly without much change in their activity.

It is useful to identify and use a small number of features rather than a larger number. In this paper, we have been able to identify a handful of features that seem to work for several, diverse data sets. This is important as the size of these archives increase significantly over time.

8. CONCLUSIONS

In this work, based on the dynamic nature of CQA services, we have taken human behavior into account and have translated them into a set of temporal features for predicting answer quality in CQA services. For these services, capturing user behavior/characteristics is important which was lacking in the traditional features proposed in the literature (both textual and non-textual). In fact, temporal features can be viewed as a generalization of some of the traditional features. Further, we have demonstrated the effectiveness and superiority of a few (4 to 5) temporal and non-temporal features by comparing our features with the large number of features (21) and the classification approach used in the literature on multiple diverse data sets. To the best of our knowledge, this is the first time temporal features are proposed/used for answer quality prediction (although they have been used in other applications).

We have also argued for ranking of *all* answers rather than classifying an answer merely as a best and non-best answer. For this purpose, we argued for the use of learning to rank approaches as a more appropriate model for predicting accuracy of answer quality as it pertains to CQA services. Using the RankSVM learning to rank approach, we have performed extensive experimental analysis on diverse data sets to demonstrate that the proposed features work well for predicting the *best* answer as well as *non-best* answer quality.

As we have noticed in the TT data set, effectiveness of temporal features depends on the dynamic nature of the data set. Hence, it is important to infer how dynamic the data set is from the characteristics of the data set itself. This also helps in identifying new features that are based on this aspect of the data set. We are currently exploring this by analyzing more data sets.

In this paper, we have used the best answer identified in the service for training as well as validation. It will also be useful to do an independent manual baseline and compare the performance of features proposed in this paper. This will allow us to generalize the effectiveness of this approach to a large class of data sets. We are pursuing this using the Amazon Mechanical Turk for this purpose.

Our future work also builds upon this framework to predict expertise of users – both globally and concept-wise. This will allow CQA services to identify their expert users and offer them incentives to improve user experience. This can also be used for coverage of topics as well as identifying best answers from archives.

REFERENCES

Bian, J., Liu, Y., Agichtein, E., and Zha, H. 2008. Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media. In *WWW*. ACM, Madrid, Spain, 467–476.

Brin, S. and Page, L. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems 30,* 1-7, 107–117.

Campbell, C. S., Maglio, P. P., Cozzi, A., and Dom, B. 2003. Expertise Identification Using Email Communications. In *CIKM*. ACM, New Orleans, Louisiana, USA, 528–531.

Cho, J. and Adams, R. E. 2005. Page Quality: In Search of an Unbiased Web Ranking. In *SIGMOD*. ACM, Baltimore, Maryland, USA, 551–562.

Dom, B., Eiron, I., Cozzi, A., and Zhang, Y. 2003. Graph-based Ranking Algorithms for E-mail Expertise Analysis. In *SIGMOD Workshop*. ACM, San Diego, California, USA, 42–48.

Everitt, B. S. and Skrondal, A. 2006. *The Cambridge Dictionary of Statistics (Second Edition)*. Vol. 4. Cambridge University Press Cambridge.

Harper, F. M., Raban, D., Rafaeli, S., and Konstan, J. A. 2008. Predictors of Answer Quality in Online Q&A Sites. In *SIGCHI*. ACM, Florence, Italy, 865–874.

Jeon, J., Croft, W. B., Lee, J. H., and Park, S. 2006. A Framework to Predict the Quality of Answers with Non-textual Features. In *SIGIR*. ACM, Seattle, Washington, USA, 228–235.

Joachims, T. 2002. Optimizing Search Engines Using Clickthrough Data. In *SIGKDD*. ACM, Edmonton, Alberta, Canada, 133–142.

Jurczyk, P. and Agichtein, E. 2007. Discovering Authorities in Question Answer Communities by Using Link Analysis. In *CIKM*. ACM, Lisboa, Portugal, 919–922.

Kendall, M. G. 1938. A New Measure of Rank Correlation. *Biometrika 30,* 1/2, 81–93.

Kleinberg, J. M. 1999. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM 46,* 5, 604–632.

Littlepage, G. E. and Mueller, A. L. 1997. Recognition and Utilization of Expertise in Problem-solving Groups: Expert Characteristics and Behavior. *Group Dynamics: Theory, Research, and Practice 1,* 4, 324–328.

Liu, Y., Bian, J., and Agichtein, E. 2008. Predicting Information Seeker Satisfaction in Community Question Answering. In *SIGIR*. ACM, Singapore, 483–490.

Page, L., Brin, S., Motwani, R., and Winograd, T. 1999. The PageRank Citation Ranking: Bringing Order to the Web. Tech. rep., University of Stanford, California, USA. `http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf`.

Radev, D. R., Qi, H., Wu, H., and Fan, W. 2002. Evaluating Web-based Question Answering Systems. In *LREC*. Vol. 1001. European Language Resources Association, Las Palmas, Canary Islands, Spain, 109–112.

Salton, G. M., Wong, A., and Yang, C. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM 18,* 11, 613–620.

Shah, C. and Pomerantz, J. 2010. Evaluating and Predicting Answer Quality in Community QA. In *SIGIR*. ACM, Geneva, Switzerland, 411–418.

Strong, D. M., Lee, Y. W., and Wang, R. Y. 1997. Data Quality in Context. *Communications of the ACM 40,* 5, 103–110.

Surdeanu, M., Ciaramita, M., and Zaragoza, H. 2008. Learning to Rank Answers on Large Online QA Collections. In *ACL*. The Association for Computer Linguistics, Columbus, Ohio, USA, 719–727.

Zhang, J., Ackerman, M. S., and Adamic, L. 2007. Expertise Networks in Online Communities: Structure and Algorithms. In *WWW*. ACM, Banff, Alberta, Canada, 221–230.

Zhu, X. and Gauch, S. 2000. Incorporating Quality Metrics in Centralized/distributed Information Retrieval on the World Wide Web. In *SIGIR*. ACM, Athens, Greece, 288–295.

Yuanzhe Cai received the bachelor's degree in computer science from Computer Science, Xidian University, China, the master's degree in computer sciencefrom Renmin University in 2008, and is currently working toward the PhD degree in the Department of Computer Science and Engineering at the University ofTexas at Arlington. His research interests include ranking, data mining, and social network analysis.

Sharma Chakravarthy is Professor of Computer Science and Engineering Department at The University of Texas at Arlington (UTA). He is the founder of the Information Technology laboratory (IT Lab) at UTA. His research has spanned semantic and multiple query optimization, complex event processing, social network analysis, and web databases. He is the co-author of the book: Stream Data Processing: A Quality of Service Perspective (2009). His current research includes social network analysis, information integration, web databases, recommendation systems, integration of stream and complex event processing, and knowledge discovery. He is an ACM Distinguished Scientist and a Senior member of IEEE. He has published over 160 papers in refereed international journals and conference proceedings.