Analyzing and Predicting Not-Answered Questions in Community-Based Question Answering Services

Lichun Yang,† Shenghua Bao,‡ Qingliang Lin,†
Xian Wu, ‡ Dingyi Han,† Zhong Su,‡ and Yong Yu†
† Shanghai Jiao Tong University
{lichunyang, qllin, handy, yyu}@apex.sjtu.edu.cn
‡ IBM Research China
{baoshhua, wuxian, suzhong}@cn.ibm.com

Abstract

This paper focuses on analyzing and predicting not-answered questions in Community based Question Answering (CQA) services, such as Yahoo! Answers. In CQA, users express their information needs by submitting questions and await answers from other users. One of the key problems of this pattern is that sometimes no one helps to give answers. In this paper, we analyze the not-answered questions and give a first try of predicting whether questions will receive answers. More specifically, we first analyze the questions of Yahoo! Answers based on the features selected from different perspectives. Then, we formalize the prediction problem as supervised learning task and leverage the proposed features to make predictions. Extensive experiments are made on 76,251 questions collected from Yahoo! Answers.

Introduction

Web search engines have achieved great success in helping users find information. Generally, users submit keyword queries to express what they need and receive relevant web pages. Since search engines cannot catch exact needs, users sometimes have to go through numbers of web pages. These years, Community based Question Answering (CQA) has been proved to be a promising alternative. Take Yahoo! Answers for example, more than 80 million questions have been resolved by the community. The typical pattern of CQA is that users express what they need via natural language questions, and then await them to be answered by other users. Since humans may understand the natural language well and so forth exact needs, they may provide specific answers.

However, one of the key problems of having questions answered by other users is that sometimes no one helps to give answers. For example, according to our crawled data, over 1 out of 8 questions get no answers in Yahoo! Answers. After all, to answer questions itself is not an attractive job.

In this paper, we propose to analyze not-answered questions and to predict whether questions will receive answers in CQA services. This task is quite meaningful for askers. Needless to say that it would be very disappointing to submit a question, to wait for several hours, even days, and to end up with no answers. If we could inform the asker instantly that the posted question is not likely to get answers or even give suggestions for the asker to re-submit the question in a better way, it certainly would help a lot.

To analyze and catch the main characteristics of not-answered questions, we propose some features from different perspectives. For content features, we leverage supervised topic model (Blei and McAuliffe 2007) to discover the latent topics. Additionally, some heuristic features, such as question length, submission time and asker's profile, are also studied. Then, the prediction problem is modeled as a traditional supervised learning task. While classic learning algorithms can all be employed to solve the problem, our focus lies in discovering and analyzing the specific features that are suitable for this task.

In the experiment, we crawled a set of questions from Yahoo! Answers. There are 76251 questions in all, among which, 10424 (about 13.67%) questions get no answers. We first analyze the topic features extracted by supervised topic modeling. Then, we show the effects of the heuristic features and analyze the reasons of their affection. Meanwhile, we try to give suggestions for askers to better submit their questions. Finally, the results of prediction are provided. Classification based on our proposed features achieves much performance gain over simple word-based methods.

Related Work

Community based Question Answering (CQA) has attracted much research interest. Here, we categorize some most related studies as question search, answer finding, expert discovery and some interesting tasks.

Question search is to find similar solved questions. Jeon et al. (2005) proposed an approach to estimate the question semantic similarity based on their answers. Wang et al. (2009) studied the syntactic tree matching method to find similar questions. Answer finding is to find the right answers. Extensive research on question answering has been done on TREC data (Voorhees 2004). Quite differently, Liu et al. proposed to summarize multiple answers and provide better information to users (Liu et al. 2008c). In expert discovery, the task is to identify authorized users in communities. Zhang et al. (2007) analyzed data from Java forum and showed that link-based metrics could be powerful tools for measuring the expertise of users. Dom and Paranjpe (2008) proposed a Bayesian-based approach to estimate users' cred-

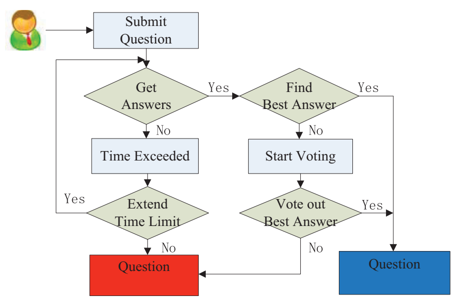

Figure 1: The questioning process in Yahoo! Answers

ibility. Moreover, there is some research work that explores interesting tasks specific in CQA. In (Li et al. 2008b), the authors pointed out one of the typical questioning phenomenon that there were many subjective questions, and proposed a co-training framework for classification in (Li et al. 2008a). Guo et al. (2008) proposed to recommend questions to possible answerers in order to improve users' participation rates in answering. In (Rodrigues et al. 2009), the authors analyzed the specific intent of users.

This work is most related to (Liu et al. 2008a; 2008b), where the authors studied the success in question answering and the information seekers' satisfaction problem in CQA. We conclude that there are two main causes that make askers unsatisfied: 1. *Unsatisfactory answer*, which is well studied in (Liu et al. 2008a; 2008b). 2. *No answer*, which is explored in this paper. In the experiment of (Liu et al. 2008a; 2008b), the authors omitted not-answered questions. Hence, their unsatisfactory analysis is limited to cause 1, i.e. unsatisfactory answer. Additionally, it is meaningful to distinguish the causes. For cause 1, we should suggest user to ask expert for help; while for cause 2, we should suggest user to modify his/her question. To the best of our knowledge, we are the first to analyze the not-answered questions in CQA services.

Not-Answered Questions

The Process of Question and Answering

Figure 1 shows the Yahoo! Answers' policy of question and answering. After a user submits a question, he/she may start to receive answers. Whenever there are some answers, the user may choose a certain one to be the *best answer*. This will turn the question to be *resolved* and end the question. If the user is not certain which answer is the best or the user does not make choices, the question will come into a vote for the best answer. Note that there will be a *no best answer choice*. If there exist answers but most users choose *no best answer*, the question will be deleted.

In Yahoo! Answers, there is a time limit for answering questions. After a question submission, there is a 4-day time frame. When this time is up, the user may choose to extend the limit and get another 4-day limit. The extension can be made only once. When the question expires again, if the question still gets no answers (or there are answers but most users choose no best answer), the question will also be deleted.

Not-Answered Question Definition

In practical, there is no way to tell if a question will not receive any answer *forever*. In this paper, we make our definition according to Yahoo! Answers' policy.

Definition: "Not-Answered Questions" are the questions that expired with no best answer and have been deleted, while *"Answered Questions"* are the resolved questions.

It is reasonable to set a time limit and delete questions that expire with no best answers. Firstly, if a question is not answered in the early several days after its submission, it is not likely to get answers in the future. Secondly, even if there are some answers after several days, they are not likely to be helpful for the original question asker, since much time has passed and the asker may have already got answers in another way or the question maybe invalid at that time.

Features to Analyze

Content Features

We choose to apply statistical topic learning methods to extract content features, instead of natural language processing based approaches. We will use topics as features instead of words, which effectively reduces the feature dimension. More specifically, the proportions of words from every topic are taken as the features. In order to extract topics that may distinguish not-answered questions, supervised Latent Dirichlet Allocation (SLDA) (Blei and McAuliffe 2007) is employed. In SLDA, each document is paired with a label, and the labels are modeled as generated from the latent topics. Thus, topic discovery is guided by the labels and the extracted topics would be more predictive. The modeling details can be found in (Blei and McAuliffe 2007).

Heuristic Features

Besides question contents, there is other information of questions that may help to distinguish not-answered questions. Here, we propose 7 heuristic features.

1. *Question length*: The length of a document might be always the first to come up. For the prediction task proposed here, question length somehow reflects whether the question is explained in detail, thus it may be predictive. Here, we use the number of words as question length.

2. *Category*: In Yahoo! Answers, there is a well-structured taxonomy for the questions. The most common way to browse questions is to select a category first. Thus, the questions in a same category may attract the same group of users, and may have similar patterns in being answered or not. We use both the top categories and the secondary categories of a question as features.

3. *Category matching*: Following category feature, we consider the category matching problem. We use simple cosine similarity between the question and its category (using words as dimensions) to describe whether it has been put in a suitable category. Both top and secondary category matching is studied.

4. *Asker history*: Experienced users are more familiar with CQA service as well as the community and may be aware of what topics are popular; while newcomers probably wonder about what to ask or how to ask. Three features about the

asker, the number of resolved questions asked, answers and best answers, are employed.

5. *Question time*: The online user number varies with day hours. The questions posted at evening, when there are quite a lot of users online, and those posted early in the morning, when most people get offline, may have different chances to get answered. We use the day hour as a feature.

6. *Question subjectivity*: One of the key characteristics of CQA is that there are quite many subjective questions (Li et al. 2008b). Subjective questions are those asking for answers with private states, e.g. personal interests, opinions, judgments; on the other hand, objective questions require authoritative information (Li et al. 2008a). Instead of classifying the questions and getting a binary label (Li et al. 2008a), we propose to simply count the number of subjective words in a question to estimate subjectivity. The subjective word list is from (Wilson et al. 2005).

7. *Polite words*: We also try to see whether polite words in the questions may help to attract answerers, and thus, make them more likely to be answered. 6 words that show politeness are selected manually, i.e. “thank”, “thanks”, “please”, “could”, “would” and “help”. The total number of these words in a question is the feature.

Prediction Problem

Problem Definition

Formally, we model the prediction problem as a traditional supervised learning binary classification problem.

Definition: “Not-Answered Question Prediction”. Given a training set of expired questions, the prediction task is to explore the newly posted questions and predict whether they will be answered or not.

Classification Algorithms

We use Weka tool (Hall et al. 2009) to perform classification experiments. Four classic classification algorithms are applied. 1 *Naïve Bayes*: A simple, fast, yet surprisingly effective method. 2 *Decision Trees*: Decision Trees can handle quite different types of features. J48, the Weka implementation of the famous C4.5 algorithm, is used (Quinlan 1996). 3 *Boosting*: We also select a meta-learning algorithm. AdaBoost is employed, which has been shown to be quite effective for many tasks (Freund and Schapire 1996). 4 *SVM*: Support Vector Machines are widely used classifier. Specifically, we use Weka implementation of SMO (Platt 1998).

Our focus lies in discovering and analyzing the specific features suitable for the task. The content and heuristic features proposed in this paper are all leveraged. We compare these with simple word-based features in the experiments.

Experiment Setup

Data Preparation

In order to build a test bed for the prediction task, we need to get a set of questions containing not only answered questions but also deleted ones. Considering the representativeness of the data and the experiments, we try to crawl all the questions posted in a selected time period. In our experiments, we choose to use the questions of a whole day, more specifically, that is April 1st 2010. The questions are collected through the Yahoo! Answers API. Moreover, in order to study the user history features, we crawled the URL page of each user and got the statistics of question and answering.

Our consolidated data set contains 76,251 questions in all, among which 10,424 were deleted automatically by Yahoo! Answers. The number of related askers is 51,983.

Topic id	Top words	Prob.
47	april, fools, day, good, joke, prank, fool, easter, today, play	0.0163
42	dog, food, dogs, hit, cat, puppy, bed, eat, vet, outside	0.0294
11	eat, make, food, good, recipe, water, chicken, chocolate, eggs, dinner	0.0298
6	com, youtube, http, www, video, yahoo, email, website, account, link	0.3407
22	phone, buy, cell, mobile, use, card, new, free, ipod, iphone	0.3345
18	game, internet, play, games, download, router, use, connect, windows, ps	0.2723

Table 1: Sample topics learned from SLDA

Evaluation Metrics and Baseline Setting

As shown in the data statistics, we are dealing with an imbalanced classification problem. Moreover, the class we need to emphasis is the minority one, the not-answered questions. Therefore, we treat the F1 score of this class (denoted as F1-NAQ) as our main evaluation metric.

We use simple word-based features as the baseline. We first extract the words from the questions and remove stopwords. Then, we employ a standard feature selection method, i.e. keep 1,000 words with highest frequencies. The words are weighted using standard *tf-idf* scheme.

Experimental Results

Topic Feature Analysis

According to the scale of our dataset, we empirically set the topic number of SLDA to be 50.

Table 1 gives 6 sample topics of a 50-topic SLDA run. The column “Prob.” shows the probability of generating not-answered question inferred by SLDA. The first 3 of the topics are the most likely to be answered and the other 3 are the contrary. Interestingly, the first topic that comes up is about April Fools’ Day. The crawled questions are all from April 1st and it seems that *“April Fool”* is a very hot topic, which is related with a very low probability of becoming not-answered. The words in other topics are also quite coherent. The other low “Prob.” topics are about *pet* and *food*; and the high “Prob.” ones are about *Yahoo!, phone* and *game*. In the analysis of category feature in the next subsection, some results further verify the effectiveness and the correctness of these extracted topics and their related not-answered probability.

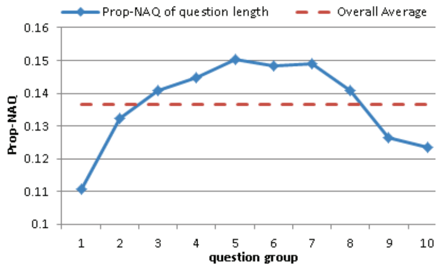

Figure 2: The question length feature

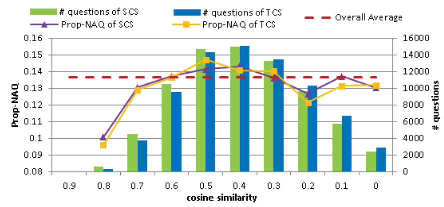

Figure 3: The category similarity feature

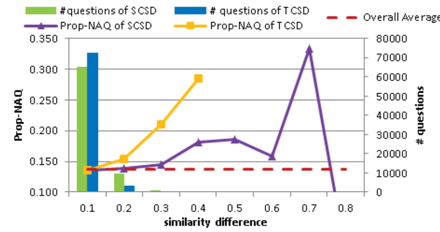

Figure 4: The similarity difference feature

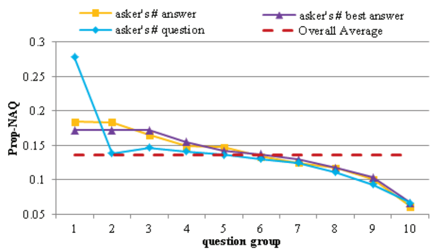

Figure 5: The user history feature

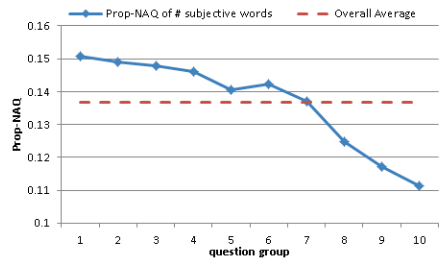

Figure 6: The subjective word feature

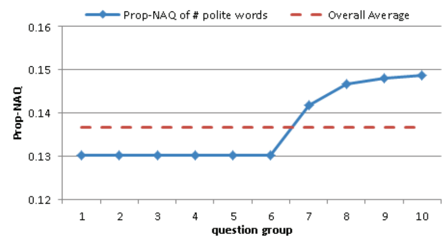

Figure 7: The polite word feature

Top Category	Prop-NAQ	normal 2nd-level #	normal 2nd-level Prop-NAQ	other 2nd-level #	other 2nd-level Prop-NAQ
Consumer Electronics	0.2789	2686	0.2837	154	0.1948
Education & Reference	0.2439	3469	0.2300	336	0.3869
Games & Recreation	0.2404	2814	0.2463	323	0.1889
Food & Drink	0.0578	733	0.0518	271	0.0738
News & Events	0.2822	255	0.2784	32	0.3125
Pets	0.0428	1647	0.0376	200	0.0850
Politics & Government	0.0762	3342	0.0742	214	0.1075
Environment	0.0625	77	0.0519	35	0.0857
Pregnancy & Parenting	0.0435	2662	0.0413	98	0.1020
Yahoo! Products	0.3806	880	0.3591	100	0.5700
Over all	0.1367	70924	0.1321	5327	0.1986

Table 2: Question numbers and Prop-NAQ across categories

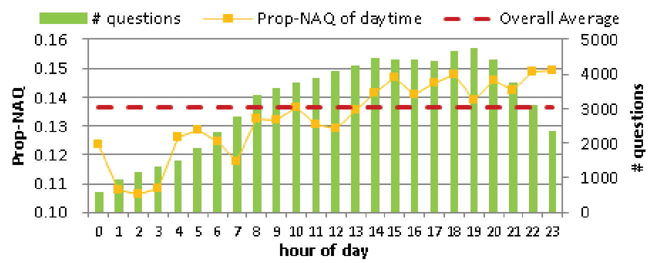

Figure 8: The question time feature

Heuristic Feature Analysis

In this subsection, we are to analyze the proposed heuristic features' ability to predict questions' answered or not-answered. We show the correlation between the features' value and the proportion of not-answered questions. In addition, we will discuss the reasons for that, and try to provide suggestions for askers.

1. *Question length*: Figure 2 shows the proportion of not-answered questions (denoted as Prop-NAQ) with respect to questions' lengths. We rank the questions according to their length in ascending order, then equally partition them into 10 groups. The left most data point stands for the Prop-NAQ of the 10% (about 7,600) questions with shortest legnths. Figures 5, 6, and 7apply similar methods to show Prop-NAQ with respect to the corresponding features. The dashed line gives the average Prop-NAQ of the whole set, which is also shown in other figures of this subsection.

For top 10% short and top 10% long questions, they get the lowest two Prop-NAQ, while medium length questions are less likely to get answered. For short questions, such as survey type questions "What color do you like most?", users can finish reading and answering the question in a very short time. On the other hand, for very long questions, such as something about science which needs detailed descriptions, users with the same interest and expertise are likely to answer it. While in the middle groups, perhaps the questions are not so interesting and unnecessarily long, making others not willing to read, thus get high Prop-NAQ.

2. *Category*: We list the statistics of 10 sample top level categories of Yahoo! Answers in Table 2. Besides Prop-NAQ of each category, we further show some statistics related to secondary categories. For each top category, there is a corresponding secondary category called *"Other - X"*, where *X* is the name of the top category. The number of questions and Prop-NAQ of both "normal" secondary categories (all except *"other"*) and the *"other"* category under each top is shown on the last 4 columns in Table 2.

The category *"Yahoo! Products"* gets the highest Prop-NAQ. Since users are not from Yahoo! company, they may be not so familiar with the Yahoo! products to provide answers. In addition, it is not likely that users would be interested in that category and always pay attention to the questions posted there. The second high Prop-NAQ category is *"News & Events"*. In fact, there are not many questions. Users may prefer to discuss this type of issues on some other web sites, especially news portals. The 2 lowest Prop-NAQ categories are *"Pregnancy & Parenting"* and *"Pets"*. We think users interested in these issues are very warmhearted and quite willing to help each other.

Cross checking with the topics extracted from SLDA, we can find that they are quite co-related. Many extracted topics fit to some categories and the related Prop-NAQ is reasonable. However, this does not mean that the two features are duplicates. The *"April Fool"* topic is a clear example, which fits in multiple categories. Other topics may also across several categories, as has been shown in (Guo et al. 2008).

Then we analyze the results of *"other"* categories. As we can see from the table, for most top categories, the questions in the corresponding "other" category has a higher Prop-NAQ. The reason is that answerers are not likely to pay much attention on those categories with miscellaneous questions. If there is a suitable secondary category, askers had better not put the question into the *"other"* category.

3. *Category matching*: In Figure 3, Prop-NAQ of questions with different cosine similarities are shown. Note that for this analysis, we do not partition the questions equally. This is because the feature value should also be discussed. The horizontal axis is cosine similarity and the questions are partitioned into 10 groups according to this value. The first group are questions with similarity 0.9-1 and so on so forth. Both questions' *Top Category Similarities* (TCS) and *Secondary Category Similarities* (SCS) are shown. Both Prop-NAQ and the number of questions of each group are shown.

For the group with similarity 0.9-1, there are no questions. Then for the next top similarity group, they have the lowest Prop-NAQ. The high similarities with the category not only make it clear that the questions are very probably to have been put in a right category, but also indicate that they are hot topics. However, as the similarity goes low, the trend is not very clear. In addition, we check how the question is matched to other categories and the effect of it. The difference between the similarity with the current category and that with the most similar category is taken as the feature. The results are shown in Figure 4 and the representations are similar with Figure 3. *SCSD* and *TCSD* in the figure represent *Secondary Category Similarity Difference* and that of top category, respectively.

The lower the similarity difference is, the more likely that the question is put in a correct category. The first group contains most questions. This indicates that most questions have been put in the right category. However, this seems to be a quite basic requirement for questions to be answered, since the group does not show a clear lower Prop-NAQ comparing to the average. As the difference increases, there is a clear trend that the questions are becoming less likely to be answered. Note that the last 3 data points of SCSD represent only 19, 3, 2 questions respectively. They probably contain too few instances to reveal statistic trend. Moreover, TCSD seems to be more useful. This is because matching the top category is an important requirement, while secondary categories start to split topics too specifically.

4. *Asker history*: We show the distinguishing ability of user history features in Figure 5. The questions are ranked according to the 3 features separately in ascending order and equally partitioned into 10 groups. Note that there are quite a lot of new comers, i.e. with no resolved questions, no answers or best answers, which are 4396, 14934 and 25435 users respectively; while one data point stands for 10% questions, that is about 7600. Therefore, the first several points are dominated by the same set of questions, which resulted in that they get the similar Prop-NAQ.

Clearly, users with more question and answering experiences are more likely to get their questions answered. The three different metrics reveal almost the same trend. One exception is the resolved question number of users. Ranked by the askers' number of resolved questions, the 10% questions with lowest number show a very high Prop-NAQ. We think this is because new comers are not so familiar with Yahoo! Answers and not able to post questions appropriately.

5. *Question time*: We split the day time into 24 groups, one for an hour, and show both the question number and Prop-NAQ of each group in Figure 8. Though more questions indicate more online users and thus more potential answerers, it would make it difficult for users to find interesting questions when there are many questions posted too frequently. Generally, it seems that the more questions, the higher Prop-NAQ. Moreover, there are some "lags" between this correlation, because answers usually come up several hours after the question submission.

6. *Question subjectivity*: Figure 6 shows Prop-NAQ of ranked question groups, sorted by number of subjective words, ascending. As is shown, we can see a clear trend that the more subjective words, the more likely that the questions will be answered. Subjective questions just require answerers to share some personal opinions, experiences or questions, if users do not know the answer beforehand, it is usually not easy to find reliable information to give answers, and users probably do not want to spend much time on this.

7. *Polite words*: We show Prop-NAQ of ranked question groups, sorted by number of polite words, ascending, in Figure 7. There are 48045 questions (over 60%) with no polite words, so the top 6 data points all relate to the same Prop-NAQ. Quite surprisingly, we find Prop-NAQ increases as the number of polite words increases. After checking some sample questions, we conclude that questions containing polite words are usually those related to the asker along, describing complicated or troublesome experiences and asking for suggestions or advices. These questions need careful reading, analyzing and thinking to give suitable answers. Though the askers usually use polite words to show his/her appreciation for help, they are still not likely to get answers.

Prediction Results

In the classification experiments, all the results are from a 10-fold cross validation run. Note that SLDA takes document labels into consideration; therefore we separately train SLDA on the 9-fold training set for each run. Weka Tool

Feature	NB	AdaBoost	C4.5	SMO
Word-based	0.259	0.211	0.263	0.272
Topic	0.288	0.253	0.294	0.317
Heuristic	0.271	0.311	0.309	0.297
Overall	0.304	0.289	0.314	0.325

Table 3: F1-NAQ results training on balanced data

is employed and the classification algorithms are all implemented and integrated in Weka (Hall et al. 2009).

When we make training on the whole set, due to the imbalance nature of the data, three algorithms, i.e. AdaBoost, C4.5 and SMO, report all questions to be “answered”. As discussed in (Provost 2000), classification on imbalanced data is generally difficult, and it is common that some classifiers would label all instances as the majority class. Only Naïve Bayes works well, since it takes prior label distribution into consideration. The F1-NAQ for word-based, topic, and heuristic features are 0.272, 0.311, 0.298, respectively. Word-based and topic features both model the contents, while topic features outperform word-based method a lot, not to mention that the feature dimension is much less. This verifies the success of SLDA that extracts predictive topics. Heuristic features also provide competitive results.

As suggested in (Provost 2000), we use a simple sampling method to deal with the imbalance problem, i.e. to keep all the not-answered questions and randomly sample the same number of answered questions for training. The classification results on this type of training sets are provided in Table 3. Though the sampling phase drops many training instances, a balanced training set makes the algorithms more effective. As shown in the table, the algorithms all work well. Additionally, since the number of instances is much less, it takes much less time to train models. Again, we get much performance gain from topic features over word-based features; and also, the heuristic features produce good results. If we combine the features together without word-based features (“Overall” row), the highest F1-NAQ is achieved.

However, current prediction results are still not good enough for practical use. This work is a first try and there still leaves much to do.

Conclusions and Future Work

Community based question answering service (CQA), such as Yahoo! Answers, has been proved to be an effective new way to find and share information. Most work has been focused on the answered questions, especially on finding and extracting information; while not-answered ones are overlooked. In order to help users avoid getting no answers, we propose to analyze the not-answered questions and give a first try on prediction.

As for future work, we will propose more features and catch more characteristics of not-answered questions. For the prediction task, we will try to derive a specific classification model and further improve the prediction results.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments and helpful suggestions.

References

Blei, D.; McAuliffe, J. 2007. Supervised topic models. In *Proc. of NIPS ’07*, pp. 121-128.

Dom, B.; and Paranjpe, D. 2008. A Bayesian Technique for Estimating the Credibility of Question Answerers. In *Proc. of SDM ’08*, pp. 399-409.

Freund, Y.; and Schapire, R. 1996. Experiments with a new boosting algorithm. In *Proc.of ICML ’96*, pp. 148-156.

Griths, T. L.; and Steyvers, M. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228-5235.

Guo, J.; Xu, S.; Bao, S.; and Yu, Y. 2008. Tapping on the potential of q&a community by recommending answer providers. In *Proc. of CIKM ’08,* pp. 921-930.

Gyongyi, Z.; Koutrika, G.; Pedersen, J.; and Garcia-Molina, H. 2008. Questioning Yahoo! Answers. First Workshop on Question Answering on the Web, held at WWW ’08.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. *The WEKA Data Mining Software: An Update. SIGKDD Explorations,* Volume 11, Issue 1.

Jeon, J.; Croft, W.; and Lee, Joon Ho. 2005. Finding Semantically Similar Questions Based on Their Answers. In *Proc. of SIGIR ’05,* pages 617-618.

Li, B.; Liu, Y.; and Agichtein, E. 2008a. CoCQA: Co-Training Over Questions and Answers with an Application to Predicting Question Subjectivity Orientation. In *Proc. of EMNLP ’08,* pp. 937-946.

Li, B.; Liu, Y.; Ram, A.; Garcia, E. V.; and Agichtein, E. 2008b. Exploring question subjectivity prediction in community QA. In *Proc. of SIGIR ’08,* pp. 735-736.

Liu, Y.; and Agichtein, E. 2008a. You’ve Got Answers: Towards Personalized Models for Predicting Success in Community Question Answering. In *Proc. of ACL ’08,* pp. 97-100.

Liu, Y.; Bian, J.; and Agichtein, E. 2008b. Predicting information seeker satisfaction in community question answering. In *Proc. of SIGIR ’08,* pp. 483-490.

Liu, Y.; Li, S.; Cao, Y.; Lin, C.; Han, D.; and Yu, Y. 2008c. Understanding and summarizing answers in community-based question answering services. In *Proc. of Coling ’08,* pp. 497-504.

Platt, J. C. 1998. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel MethodsSupport Vector Learning,* pages185208.

Provost, F. 2000. Machine Learning from Imbalanced Data Sets. AAAI Workshop on Imbalanced Data Sets, 2000.

Quinlan, J. 1996. Improved use of continuous attributes in c4.5. In *Journal of Articial Intelligence Research.*

Rodrigues, E. M.; and Milic-Frayling, N. 2009. Socializing or Knowledge sharing? Characterizing Social Intent in Community Question Answering. In *Proc. of CIKM ’09,* pp. 1127-1136.

Wang, K.; Ming, Z.; and Chua, T. 2009. A syntactic tree matching approach to finding similar questions in community-based qa services. In *Proc. of SIGIR ’09,* pp. 187-194.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. of HLT/EMNLP ’05.*

Voorhees, E. M. 2004. Overview of the TREC 2004 question answering track. In *Proc. of TREC’04.*

Zhang, J.; Ackerman, M. S.; and Adamic, L. 2007. Expertise Networks in Online Communities: Structure and Algorithms. In *Proc. of WWW ’07,* pp. 221-230.