You’ve Got Answers: Towards Personalized Models for Predicting Success in Community Question Answering

Yandong Liu and Eugene Agichtein
Emory University
{yliu49,eugene}@mathcs.emory.edu

Abstract

Question answering communities such as Yahoo! Answers have emerged as a popular alternative to general-purpose web search. By directly interacting with other participants, information seekers can obtain specific answers to their questions. However, user success in obtaining satisfactory answers varies greatly. We hypothesize that satisfaction with the contributed answers is largely determined by the asker’s prior experience, expectations, and personal preferences. Hence, we begin to develop personalized models of asker satisfaction to predict whether a particular question author will be satisfied with the answers contributed by the community participants. We formalize this problem, and explore a variety of content, structure, and interaction features for this task using standard machine learning techniques. Our experimental evaluation over thousands of real questions indicates that indeed it is beneficial to personalize satisfaction predictions when sufficient prior user history exists, significantly improving accuracy over a “one-size-fits-all” prediction model.

1 Introduction

Community Question Answering (CQA) has recently become a viable method for seeking information online. As an alternative to using general-purpose web search engines, information seekers now have an option to post their questions (often complex, specific, and subjective) on Community QA sites such as Yahoo! Answers, and have their questions answered by other users. Hundreds of millions of answers have already been posted for tens of millions of questions in Yahoo! Answers. However, the success of obtaining satisfactory answers in the available CQA portals varies greatly. In many cases, the questions posted by askers go un-answered, or are answered poorly, never obtaining a satisfactory answer.

In our recent work (Liu et al., 2008) we have introduced a general model for predicting asker satisfaction in community question answering. We found that previous asker history is a significant factor that correlates with satisfaction. We hypothesize that asker’s satisfaction with contributed answers is largely determined by the asker expectations, prior knowledge and previous experience with using the CQA site. Therefore, in this paper we begin to explore how to personalize satisfaction prediction - that is, to attempt to predict whether a specific information seeker will be satisfied with any of the contributed answers. Our aim is to provide a “personalized” recommendation to the user that they’ve got answers that satisfy their information need.

To the best of our knowledge, ours is the first exploration of personalizing prediction of user satisfaction in complex and subjective information seeking environments. While information seeker satisfaction has been studied in ad-hoc IR context (see (Kobayashi and Takeda, 2000) for an overview), previous studies have been limited by the lack of realistic user feedback. In contrast, we deal with complex information needs and community-provided answers, trying to predict subjective ratings provided by users themselves. Furthermore, while automatic complex QA has been an active area of research, ranging from simple modification to factoid QA technique (e.g., (Soricut and Brill, 2004)) to knowledge intensive approaches for specialized domains, the technology does not yet exist to automatically answer open domain, complex, and subjective questions. Hence, this paper contributes to both the understanding of complex question answering, and explores evaluation issues in a new setting.

The rest of the paper is organized as follows. We describe the problem and our approach in Section 2, including our initial attempt at personalizing satisfaction prediction. We report results of a large-scale evaluation over thousands of real users and
tens of thousands of questions in Section 3. Our results demonstrate that when sufficient prior asker history exists, even simple personalized models result in significant improvement over a general prediction model. We discuss our findings and future work in Section 4.

2 Predicting Asker Satisfaction in CQA
We first briefly review the life of a question in a QA community. A user (the asker) posts a question by selecting a topical category (e.g., “History”), and then enters the question and, optionally, additional details. After a short delay the question appears in the respective category list of open questions. At this point, other users can answer the question, vote on other users’ answers, or interact in other ways. The asker may be notified of the answers as they are submitted, or may check the contributed answers periodically. If the asker is satisfied with any of the answers, she can choose it as best, and rate the answer by assigning stars. At that point, the question is considered as closed by asker. For more detailed treatment of user interactions in CQA see (Liu et al., 2008). If the asker rates the best answer with at least three out of five “stars”, we believe the asker is satisfied with the response. But often the asker never closes the answer personally, and instead, after a period of time, the question is closed automatically. In this case, the “best” answer may be chosen by the votes, or alternatively by automatically predicting answer quality (e.g., “thumbs up” in Yahoo! Answers), negative votes (“thumbs down”), and derived statistics such as the maximum of positive or negative votes received for any answer (e.g., to detect cases of brilliant answers or, conversely, blatant abuse).

Based on our exploration we believe that the main reasons for not “closing” a question are a) the asker loses interest in the information and b) none of the answers are satisfactory. In both cases, the QA community has failed to provide satisfactory answers in a timely manner and “lost” the asker’s interest. We consider this outcome to be “unsatisfied”. We now define asker satisfaction more precisely:

Definition 1 An asker in a QA community is considered satisfied iff: the asker personally has closed the question and rated the best answer with at least 3 “stars”. Otherwise, the asker is unsatisfied.

This definition captures a key aspect of asker satisfaction, namely that we can reliably identify when the asker is satisfied but not the converse.

2.1 Asker Satisfaction Prediction Framework
We now briefly review our ASP (Asker Satisfaction Prediction) framework that learns to classify whether a question has been satisfactorily answered, originally introduced in (Liu et al., 2008). ASP employs standard classification techniques to predict, given a question thread, whether an asker would be satisfied. A sample of features used to represent this problem is listed in Table 1. Our features are organized around the basic entities in a question answering community: questions, answers, question-answer pairs, users, and categories. In total, we developed 51 features for this task. A sample of the features used are listed in the Figure 1.

- **Question Features**: Traditional question answering features such as the wh-type of the question (e.g., “what” or “where”), and whether the question is similar to other questions in the category.
- **Question-Answer Relationship Features**: Overlap between question and answer, answer length, and number of candidate answers. We also use features such as the number of positive votes (“thumbs up” in Yahoo! Answers), negative votes (“thumbs down”), and derived statistics such as the maximum of positive or negative votes received for any answer (e.g., to detect cases of brilliant answers or, conversely, blatant abuse).
- **Asker User History**: Past asker activity history such as the most recent rating, average past satisfaction, and number of previous questions posted. Note that only the information available about the asker prior to posting the question was used.
- **Category Features**: We hypothesized that user behavior (and asker satisfaction) varies by topical question category, as recently shown in reference (Agichtein et al., 2008). Therefore we model the prior of asker satisfaction for the category, such as the average asker rating (satisfaction).
- **Text Features**: We also include word unigrams and bigrams to represent the text of the question subject, question detail, and the answer content. Separate feature spaces were used for each attribute to keep answer text distinct from question text, with frequency-based filtering.

**Classification Algorithms**: We experimented with a variety of classifiers in the Weka framework (Witten and Frank, 2005). In particular, we compared Support Vector Machines, Decision trees, and Boosting-based classifiers. SVM performed the best
of the three during development, so we report results using SVM for all the subsequent experiments.

### 2.2 Personalizing Asker Satisfaction Prediction

We now describe our initial attempt at personalizing the ASP framework described above to each asker:

- **ASP_Pers+Text**: We first consider the naive personalization approach where we train a separate classifier for each user. That is, to predict a particular asker’s satisfaction with the provided answers, we apply the individual classifier trained solely on the questions (and satisfaction labels) provided in the past by that user.

- **ASP_Group**: A more robust approach is to train a classifier on the questions from the group of users similar to each other. Our current grouping was done simply by the number of questions posted, essentially grouping users with similar levels of “activity”. As we will show below, text features only help for users with at least 20 previous questions. So, we only include text features for groups of users with at least 20 questions.

Certainly, more sophisticated personalization models and user clustering methods could be devised. However, as we show next, even the simple models described above prove surprisingly effective.

### 3 Experimental Evaluation

We want to predict, for a given user and their current question whether the user will be satisfied, according to our definition in Section 2. In other words, our “truth” labels are based on the rating subsequently given to the best answer by the asker herself. It is usually more valuable to correctly predict whether a user is satisfied (e.g., to notify a user of success). Hence, we focus on the Precision, Recall, and F1 values for the *satisfied* class.

#### Datasets: Our data was based on a snapshot of Yahoo! Answers crawled in early 2008, containing 216,170 questions posted in 100 topical categories by 158,515 askers, with associated 1,963,615 answers in total. More detailed statistics, arranged by the number of questions posted by each asker are reported in (Table 2). The askers with only one question (i.e., no prior history) dominate the dataset, as many users try the service once and never come back. However, for personalized satisfaction, at least some prior history is needed. Therefore, in this early version of our work, we focus on users who have posted at least 2 questions - i.e., have the minimal history of at least one prior question. In the future, we plan to address the “cold start” problem of predicting satisfaction of new users.

#### Methods compared:

- **ASP**: A “one-size-fits-all” satisfaction predictor that is trained on 10,000 randomly sampled questions with only non-textual features (Section 2.1).
- **ASP+Text**: The ASP classifier with text features.
- **ASP_Pers+Text** and **ASP_Group**: A personalized classifiers described in Section 2.2.

#### 3.1 Experimental Results

Figure 1 reports the satisfaction prediction accuracy for ASP, ASP_Text, ASP_Pers+Text, and ASP_Group for groups of askers with varying number of previous questions posted. Surprisingly, for ASP_Text, textual features only become helpful for users with more than 20 or 30 previous questions posted and degrade performance otherwise. Also note that baseline ASP classifier is not able to achieve higher accuracy even for users with large amount of past history. In contrast, the ASP_Pers+Text classifier, trained only on the past question(s) of each user, achieves surprisingly good accuracy – often significantly outperforming the ASP and ASP_Text classifiers. The improvement is especially dramatic for users with at least

| Feature | Description |
|---------|-------------|
| Q: Q_punctuation_density | Ratio of punctuation to words in the question |
| Q: Q_KL_div_wikipedia | KL divergence with Wikipedia corpus |
| Q: Q_KL_div_category | KL divergence with "satisfied" questions in category |
| Q: Q_KL_div_TREC | KL divergence with TREC questions corpus |

### Table 1: Sample features: Question (Q), Question-Answer Relationship (QA), Asker history (UH), and Category (CA).

### Table 2: Distribution of questions, answers and askers

| #Questions per Asker | # Questions | # Answers | # Users |
|----------------------|-------------|-----------|---------|
| 1                    | 12,279      | 1,197,089 | 158,279 |
| 2                    | 31,692      | 287,681   | 213,507 |
| 3-4                  | 23,296      | 213,507   | 158,515 |
| 5-9                  | 15,813      | 143,483   | 158,515 |
| 10-14                | 5,554       | 54,781    | 93      |
| 15-19                | 2,304       | 21,835    | 49      |
| 20-29                | 2,226       | 23,729    | 93      |
| 30-49                | 1,866       | 16,982    | 49      |
| 50-100               | 842         | 4,528     | 14      |

**Total:** 216,170, 1,963,615, 158,515

...
20 previous questions. Interestingly, the simple strategy of grouping users by number of previous questions (ASP_Group) is even more effective, resulting in accuracy higher than both other methods for users with moderate amount of history. Finally, for users with only 2 questions total (that is, only 1 previous question posted) the performance of ASP_Pers+Text is surprisingly high. We found that the classifier simply “memorizes” the outcome of the only available previous question, and uses it to predict the rating of the current question.

To better understand the improvement of personalized models, we report the most significant features, sorted by Information Gain (IG), for three sample ASP_Pers+Text models (Table 3). Interestingly, whereas for Pers 1 and Pers 2, textual features such as “good luck” in the answer are significant, for Pers 3 non-textual features are most significant.

We also report the top 10 features with the highest information gain for the ASP and ASP_Group models (Table 4). Interestingly, while asker’s average previous rating is the top feature for ASP, the length of membership of the asker is the most important feature for ASP_Group, perhaps allowing the classifier to distinguish more expert users from the active newbies. In summary, we have demonstrated promising preliminary results on personalizing satisfaction prediction even with relatively simple personalization models.

![Figure 1: Precision, Recall, and F1 of ASP, ASP_Text, ASP_Pers+Text, and ASP_Group for predicting satisfaction of askers with varying number of questions](image)

Table 3: Top 10 features by Information Gain for three sample ASP_Pers+Text models

| Feature                                      | IG     |
|----------------------------------------------|--------|
| “would” in answer                            | 0.047222|
| “answer” in question                         | 0.047222|
| “in the” in question                         | 0.047222|
| “make” in question                           | 0.047222|
| “in” in answer                               | 0.047222|
| “those” in question                          | 0.047222|
| “me” in answer                               | 0.047222|
| “anybody” in question                        | 0.047222|
| “be” in answer                               | 0.047222|
| “want” in question                           | 0.047222|
| “questions_resolved”                         | 0.047222|
| “Content_ANI_wikipedia”                      | 0.047222|
| “total_answers_received”                     | 0.047222|
| “asker_all_cat”                              | 0.047222|
| “rev_avg_rating”                             | 0.047222|
| “CA_History”                                 | 0.047222|
| “content_type_density”                       | 0.047222|

Table 4: Top 10 features by information gain for ASP (trained for all askers) and ASP_Group (trained for the group of askers with 20 to 29 questions)

| Feature                                      | ASP    | ASP_Group |
|----------------------------------------------|--------|-----------|
| “would” in answer                            | 0.010417| 0.104117  |
| “answer” in question                         | 0.010217| 0.30984   |
| “in the” in question                         | 0.004722| 0.22556   |
| “be” in answer                               | 0.004722| 0.15237   |
| “those” in question                          | 0.004722| 0.09314   |
| “make” in question                           | 0.004722| 0.09314   |
| “anybody” in question                        | 0.004722| 0.09314   |
| “the” in question                            | 0.004722| 0.09314   |
| “CA_History”                                 | 0.004722| 0.09314   |
| “Content_ANI_wikipedia”                      | 0.004722| 0.09314   |
| “total_answers_received”                     | 0.004722| 0.09314   |

4 Conclusions

We have presented preliminary results on personalizing satisfaction prediction, demonstrating significant accuracy improvements over a “one-size-fits-all” satisfaction prediction model. In the future we plan to explore the personalization more deeply following the rich work in recommender systems and collaborative filtering, with the key difference that the asker satisfaction, and each question, are unique (instead of shared items such as movies). In summary, our work opens a promising direction towards modeling personalized user intent, expectations, and satisfaction.

References

E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. 2008. Finding high-quality content in social media with an application to community-based question answering. In Proceedings of WSDM.

J. Jeon, W.B. Croft, J.H. Lee, and S. Park. 2006. A framework to predict the quality of answers with non-textual features. In Proceedings of SIGIR.

Mei Kobayashi and Koichi Takeda. 2000. Information retrieval on the web. *ACM Computing Surveys*, 32(2).

Y. Liu, J. Bian, and E. Agichtein. 2008. Predicting information seeker satisfaction in community question answering. In Proceedings of SIGIR.

R. Soricut and E. Brill. 2004. Automatic question answering: Beyond the factoid. In *HLT-NAACL*.

I. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufman, 2nd edition.