Routing Questions to Appropriate Answerers in Community Question Answering Services

Baichuan Li and Irwin King
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{bcli,king}@cse.cuhk.edu.hk

ABSTRACT

Community Question Answering (CQA) service provides a platform for increasing number of users to ask and answer for their own needs but unanswered questions still exist within a fixed period. To address this, the paper aims to route questions to the right answerers who have a top rank in accordance of their previous answering performance. In order to rank the answerers, we propose a framework called *Question Routing* (QR) which consists of four phases: (1) performance profiling, (2) expertise estimation, (3) availability estimation, and (4) answerer ranking. Applying the framework, we conduct experiments with Yahoo! Answers[1] dataset and the results demonstrate that on average each of 1,713 testing questions obtains at least one answer if it is routed to the top 20 ranked answerers.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering, selection process*

General Terms

Algorithms, Experimentation

Keywords

community question answering, question routing

1. INTRODUCTION

Community Question Answering (CQA) service is a special kind of Question Answering (QA) service which allows registered users to answer the questions asked by other people. CQA portals such as Yahoo! Answers and Baidu Zhidao[2] have attracted increasing number of users over the last few years. According to Yahoo! Answers blog[3], it has 200 million users worldwide and around 15 million visits daily.

Since CQA portals are so popular, one interesting and important question is whether this service can solve askers' questions efficiently. In order to investigate the answer to this question, we randomly track 3,000 newly posted questions in Yahoo! Answers and Baidu Zhidao respectively to observe these questions' states after two days. We find that in Yahoo! Answers only 17.6% of questions receive satisfied answers within 48 hours. For those unresolved questions, nearly 1/5 of them receive no response. For Baidu Zhidao, 22.7% of questions are well resolved. However, 42.8% of unresolved questions receive no response at all. The observations show that above two popular CAQ portals cannot solve users' questions efficiently.

In order to address this problem, we propose the framework of *Question Routing* (QR) in CQA. This framework routes new posted questions to users who are most likely to give answers in a short period. The concept of QR contains two meanings: finding (1) the "right" users who can provide high quality answers; (2) the users who receive the routed question must be able to give response quickly, i.e., they are available to answer this question in time.

Liu et al.'s work [2] which "identify the group of 'experts' who are likely to provide answers to given questions" and Qu et al.'s question recommendation [4] are similar to our work. However, answer quality and user availability were not considered in these papers. Recently, Horowitz et al. [1] developed a social search engine called "Aardvark" which "routes the question to the person in the user's extended social network most likely to be able to answer that question," where the extended social network includes popular social network websites such as Facebook[4] and LinkedIn[5]. Compared with "Aardvark", our work focus on the community in CQA services rather than the social network of the asker.

The paper is organized as follows. Section 2 details the framework of QR and several models within the framework. In Section 3, we describe the experimental setup and the analyses of results. Conclusions are given in Section 4.

2. QUESTION ROUTING IN CQA SERVICES

The whole process of QR is illustrated in Fig. 1. For a question to be routed, we first extract all answerers in the portal and build their answering performance profiles.

[1]http://answers.yahoo.com/
[2]http://zhidao.baidu.com/
[3]http://yanswersblog.com/
[4]http://www.facebook.com/
[5]http://www.linkedin.com/

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

This step is called answer performance profiling. Then we estimate each candidate's expertise on the routed question based on his/her performance profile. Finally we rank all the answerers based on their expertise.

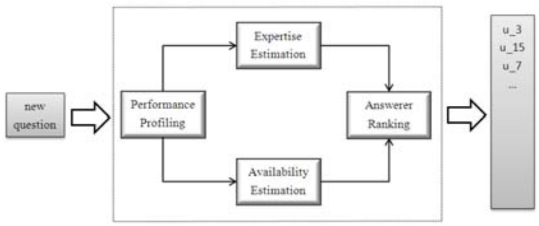


Figure 1: The framework of *Question Routing.*

2.1 Performance Profiling

In the phase of performance profiling, we establish each answerer's performance profile from his/her answering history. For the user who has answered at least one question in CQA services, we use all questions he/she once answered and the corresponding answers providing by the user to build the his/her performance profile.

2.2 Expertise Estimation

In this section we present three approaches to estimate each answerer's expertise on q_r. In the following, we use $E(u_i, q_r)$ to denote user u_i's expertise on the new question q_r, which ranges from 0 to 1. The higher value $E(u_i, q_r)$ is, the higher expertise the user u_i has for the question q_r.

2.2.1 Expertise estimation without answer quality

We adopt the query likelihood language (QLL) model as our first model. Formally, Let q_{u_i} denote all previously answered questions by user u_i. For a new question q_r, u_i's expertise on q_r is defined as how likely q_r can be generated from q_{u_i}:

$$E(u_i, q_r) = P(q_r|q_{u_i}), \tag{1}$$

$$P(q_r|q_{u_i}) = \prod_{\omega \in q_r} P(\omega|q_{u_i}). \tag{2}$$

With Jelinek-Mercer smoothing [8],

$$P(\omega|q_{u_i}) = (1-\lambda)P(\omega|q_{u_i}) + \lambda P(\omega|C), \tag{3}$$

$$P(\omega|q_{u_i}) = \frac{tf(\omega, q_{u_i})}{\sum_{\omega' \in q_{u_i}} tf(\omega', q_{u_i})}, \tag{4}$$

$$P(\omega|C) = \frac{tf(\omega, C)}{\sum_{\omega' \in C} tf(\omega', C)}, \tag{5}$$

where C is the collection of all questions and λ is a weighting coefficient to adjust the weight of smoothing, $tf(\omega, q_{u_i})$ means the term frequency of the term ω in q_{u_i} and $tf(\omega, C)$ is the term frequency of the term ω in C. We set $\lambda = 0.8$ in our experiments according to the empirical value [8].

2.2.2 Expertise estimation with answer quality

The above model assumes that the user has high expertise on new question q_r if he/she has answered many similar questions before. However, it does not consider the quality of previous answers. A user may answers a great number of questions which are similar to q_r, but we cannot reach the conclusion that the user must be an expert for question q_r if most previous answers are of low qualities. In order to obtain a more accurate prediction, we utilize user's answer quality in expertise estimation. Thus,

$$E(u_i, q_r) = \alpha \cdot P(q_r|q_{u_i}) + (1-\alpha) \cdot Q(u_i, q_r), \tag{6}$$

where $Q(u_i, q_r)$ reflects user u_i's answer quality for question q_r and α is a weighting coefficient.

We propose two models to estimate $Q(u_i, q_r)$ from previous answers' qualities of user u_i. The **Basic Model** is straightforward: it assumes the user's answer quality on new question q_r is the weighted average answers qualities of similar questions he/she answered previously. It is defined as:

$$Q_{BM}(u_i, q_r) = \frac{\sum_{q_j \sim u_i} Q(u_i, q_j) \cdot sim(q_j, q_r)}{\sum_{q_j \sim u_i} sim(q_j, q_r)}, \tag{7}$$

where $q_j \sim u_i$ denotes the questions u_i has answered and $sim(q_j, q_r)$ means the cosine similarity between question q_j and q_r.

We use vector space model [5] to represent each question and each term is weighted by its tf-idf value.

However, this model may suffers from data sparsity especially when some users only answered one question. In order to better utilize the known information, we borrow the idea of similarity fusion in collaborative filtering [7] which leverages other similar users' answer qualities on similar questions to smooth the **Basic Model**. In the **Smoothed Model**,

$$Q_{SM}(u_i, q_r) = \beta Q_{BM}(u_i, q_r) + (1-\beta)\frac{\sum_{u_j \in U/u_i} \sum_{q_k \sim u_j} Q(u_j, q_k) \cdot sim(Q_{u_j q_k}, Q_{u_i q_r})}{\sum_{u_j \in U/u_i} \sum_{q_k \sim u_j} sim(Q_{u_j q_k}, Q_{u_i q_r})}, \tag{8}$$

and

$$sim(Q_{u_j q_k}, Q_{u_i q_r}) = \frac{1}{\sqrt{\frac{1}{sim(u_i, u_j)^2} + \frac{1}{sim(q_k, q_r)^2}}}. \tag{9}$$

The cosine similarities between two users is calculated based on the following features for each user: number of total points the user owns, number of answers the user has provided, number of best answers the user has provided, number of questions the user has asked and number of stars the user received.

We use logistic regression to model the user's answer quality on previously answered questions. Given an answer $a(u_i, q_j)$ which is posted by u_i for question q_j, we use $\mathbf{a}_{ij}$ to denote the feature vector of $a(u_i, q_j)$. Let the probability of $a(u_i, q_j)$ being a good answer be $P(\mathbf{a}_{ij})$, then:

$$log\left(\frac{P(\mathbf{a}_{ij})}{1 - P(\mathbf{a}_{ij})}\right) = \sigma^T \mathbf{a}_{ij}, \tag{10}$$

while σ is the coefficient vector of the regression model.

The following features for each answer are extracted in training:

- Answer length
- Question-Answer length ratio
- # of answers for this question
- # of times the answer is rated up other users
- # of times the answer is rated down by other users
- The answerer's total points

- The answerer's best answer ratio

We apply feature conversion on the non-monotonic features using Kernel Density Estimation (KDE) [3].

2.3 Availability Estimation

Assuming one user is available to provide answers for the routed questions when he log on Yahoo! Answers, we aim to estimate whether the user logs on in several days after the routed question is posted. We model this problem as a typical trend analysis problem in time-series data mining and use an autoregressive model to make the forecasting. Formally, let $A(u_i, t)$ denote the probability that u_i is available at time t to answer routed question, usually t represents one specific day. In practice, we set $A(u_i, t) = 1$ when u_i posted at least one answer on the day t, otherwise $A(u_i, t) = 0$.

The autoregressive model can be represented as follows:

$$A(u_i, t) = \lambda_1 A(u_i, t-1) + ... + \lambda_p A(u_i, t-p) + \varepsilon, \quad (11)$$

where the term ε is the source of randomness and is called white noise. Given a group of training data $\{A(u_i, t), A(u_i, t-1), ..., A(u_i, t-p)\}_{i=1}^{m}$ where m is the number of users, we can estimate the value of $\lambda_1, \lambda_2, ..., \lambda_p$. Then we can apply the above model to predict the value of $A(u_i, t)$ when given $A(u_i, t-1), ..., A(u_i, t-p)$.

Thus, each answerer's availability for a period of time $T = \{t_1, ..., t_s\}$ is calculate as:

$$A(u_i, T) = 1 - \prod_{j=1}^{s}(1 - A(u_i, t_j)). \quad (12)$$

2.4 Answerer Ranking

We treat user expertise on q_r and user availability in a range of time T as independent and use their linear combination as the final QR score for each answerer:

$$QR(u_i, q_r, T) = \gamma \cdot E(u_i, q_r) + (1 - \gamma) \cdot A(u_i, T). \quad (13)$$

Then, all answerers are ranked according to their QR scores.

3. EXPERIMENTS

We want to investigate the answers to those research questions through experiments:

1. What's the influence of answer quality to the performance of QR?
2. Does the **Smoothed Model** give better answer quality estimation and the improvement of QR performance ?
3. Is it useful to estimate users' answer availabilities in QR?

3.1 Data Set and Experimental Setup

Our data set is a snapshot of resolved questions from April 6, 2010 to May 14, 2010 under the *Computers & Internet* category of Yahoo! Answers. The stopwords in question subjects and answer content are removed. In our experiments, the questions posted after May 6 are treated as new questions to be routed (Set A, testing data) and the left ones are treated as archive data (Set B). The ground truth for each question in Set A are the answerers who actually answer it. After splitting as stated above (we also remove the questions in Set A whose answerers all do not appear in Set B), Set A includes 1,713 questions, 5,403 answers and 2,891 answerers. Set B includes 17,182 questions, 48,663 answers and 16,298 answerers.

Recall that we use a logistic regression model to estimate each answerer's expertise on the questions which the user has answered. We adopt the community and the askers' choices to avoid manually labeling. Answers are labeled as "good" and "bad" as follows. For each question in Set B, the answer is labeled as a "good" answer only the following two conditions are met: (1) It is selected as the best answer; (2) It obtains more than 50% of rate-ups for all answers of the question. Meanwhile, one answer is labeled as a "bad" answer if it receives more than 50% of rate-downs for all answers of the question. As such, 2,153 "good" instances and 2,593 "bad" instances served as training data to estimate the parameters of the logistic regression model.

We set $T = 3$ according to our observation (In Set B, the max duration time for a question to receive an answer is 2.16 days). Moreover, we set $p = 3$ which means the first three days' answering records are used to estimate the user's availability in the fourth day.

We compare the QR performance of the following methods.

- **QLL**: For each question q_r in Set A, we use the **QLL** model in Section 2.2.1 to calculate the expertise on q_r for each answerer in Set B. Thus,

$$QR(u_i, q_r, T) = P(q_r | q_{u_i}).$$

- **Basic Q**: We use the **Basic Model** to calculate the answer quality and apply Eq. (6) to estimate each answerer's expertise in Set B. Thus,

$$QR(u_i, q_r, T) = \alpha \cdot P(q_r | q_{u_i}) + (1 - \alpha) \cdot Q_{BM}(u_i, q_r).$$

- **Smoothed Q**: We use the **Smoothed Model** to calculate the answer quality and apply Eq. (6) to estimate each answerer's expertise in Set B. Thus,

$$QR(u_i, q_r, T) = \alpha \cdot P(q_r | q_{u_i}) + (1 - \alpha) \cdot Q_{SM}(u_i, q_r).$$

- **QLL+AE**: We add answerer's availability to the **QLL** method. Thus,

$$QR(u_i, q_r, T) = \gamma \cdot P(q_r | q_{u_i}) + (1 - \gamma) \cdot A(u_i, T).$$

- **Basic Q+AE**: We add answerer's availability to the **Basic Q** method. Thus,

$$QR(u_i, q_r, T) = \gamma \cdot [\alpha \cdot P(q_r | q_{u_i}) + (1 - \alpha) \cdot Q_{BM}(u_i, q_r)] + (1 - \gamma) \cdot A(u_i, T).$$

- **Smoothed Q+AE**: We add answerer's availability to the **Smoothed Q** method. Thus,

$$QR(u_i, q_r, T) = \gamma \cdot [\alpha \cdot P(q_r | q_{u_i}) + (1 - \alpha) \cdot Q_{SM}(u_i, q_r)] + (1 - \gamma) \cdot A(u_i, T).$$

We adopt the Mean Reciprocal Rank (MRR) [6] as the evaluation metric to evaluate the performance of above methods.

3.2 Experiment Results

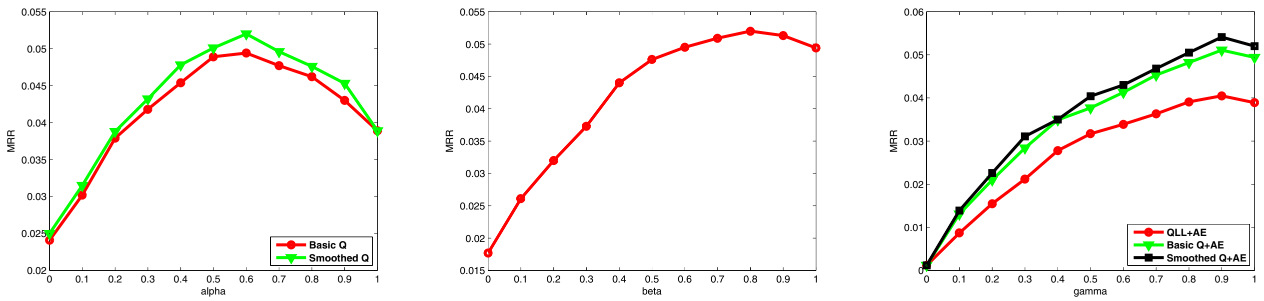
Table 1 reports each method's performance measured by MRR. Here we set $\alpha = 0.6$, $\beta = 0.8$ and $\gamma = 0.9$.

3.2.1 *The impact of answer quality*

From Table 1 we observe that utilizing users' answer qualities can improve the performance of QR significantly. The MRR values of **Basic Q** and **Smoothed Q** are 26.99% and 33.68% higher than that of **QLL** respectively. Similarly, the

Table 1: **Different methods' performance in QR**

Evaluation Metrics	**QLL**	**Basic Q**	**Smoothed Q**	**QLL+AE**	**Basic Q+AE**	**Smoothed Q+AE**
MRR	0.0389	0.0494	0.052	0.0405	0.0511	0.0541



(a) (b) (c)

Figure 2: The influence of parameter setting. (a) The MRR value of Basic Q and Smoothed Q versus various α. (b) The MRR value of Smoothed Q versus various β. (c) MRR versus γ across different methods.

MRR values of **Basic Q+AE** and **Smoothed Q +AE** are 26.17% and 33.58% higher than that of **QLL+AE**.

In order to explore the impact of α's value for QR performance, we fixed $\beta = 0.8$ and tested different settings of α and the result is reported in Fig. 2 (a). First we observe that when $\alpha = 0.6$, both **Basic Q** and **Smoothed Q** get the highest MRR. Furthermore, when $\alpha > 0.3$, the performance of them are always better than QLL. With this findings, we believe that users' answer qualities on perviously answered questions indeed provide great help for us to finding experts on the routed question.

3.2.2 *Basic Q vs. Smoothed Q*

Smoothed Q outperforms **Basic Q** since the MRR of former one is about 5% higher than that of the latter. We think it is due to the help of utilizing similar users' expertise on similar questions to smooth the user's answer quality, especially for the users who answered few questions. Figure 2 (b) give the performance of **Smoothed Q** with different settings of β ($\alpha = 0.6$), from which we find that the value of β affects the QR quality of this method to a great extent. When $\alpha = 0$ which means we just rely on similar users's expertise on similar questions to estimate the user's expertise on routed question, the MRR is much lower than no smoothing (i.e., $\alpha = 1$). The best performance of MRR is gotten when $\beta = 0.8$.

3.2.3 *The impact of answer availability estimation*

Figure 2 (c) present each method's performance when considering users' answer availability. First, routing questions based on users' answering availabilies only is very inaccurate: only about 1 out of 1,000 QRs will be successful when $\gamma = 0$. Second, **QLL+AE**, **Basic Q+AE** and **Smoothed Q+AE** perform best when γ is setting around 0.9. When $\gamma = 0.9$, the MRR of these methods are 4.11%, 3.44% and 4.04% higher than corresponding methods without availability estimation. Thus, the performance of QR can be improved by users' availability estimation.

4. CONCLUSIONS

In this paper, we introduce the concept of *Question Routing* in CQA services and propose a QR framework which considers both users' expertise and users' availabilities for providing answers in a range of time. We conduct experiments on Yahoo! Answers dataset and the results demonstrates that leveraging answer quality can greatly improve the performance of QR. In addition, utilizing similar users' answer qualities on similar questions provides a more accurate expertise estimation and thus give better QR performance. Furthermore, users' answer availability estimation can also boost the performance of QR. The best MRR value in our experiments is 0.0541, which means on average each tested question will get at least one answer if we route it to the top 20 ranked users. Considering totally there are 16,298 answerers for ranking, the result demonstrates that our QR framework has the ability to route new questions to those users who will provide answers in a short period of time.

5. ACKNOWLEDGMENTS

This work is supported by two grants from the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK4128/08E and CUHK4154/09E).

6. REFERENCES

[1] D. Horowitz and S. D. Kamvar. The anatomy of a large-scale social search engine. In *Proc. of WWW '10*, 2010.

[2] X. Liu, W. B. Croft, and M. Koll. Finding experts in community-based question-answering services. In *Proc. of CIKM '05*, 2005.

[3] J. neng Hwang, S. rong Lay, and A. Lippman. Nonparametric multivariate density estimation: A comparative study. *IEEE Trans. Signal Processing*, 42:2795–2810, 1994.

[4] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen. Probabilistic question recommendation for question answering communities. In *Proc of WWW '09*, 2009.

[5] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[6] E. Voorhees. The trec-8 question answering track report. In *TREC8*, 1999.

[7] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proc. of SIGIR '06*, 2006.

[8] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.