Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media

Effective Question Recommendation Based on Multiple Features for Question Answering Communities

Yutaka Kabutoya[†], **Tomoharu Iwata**[‡], **Hisako Shiohara**[†], and **Ko Fujimura**[†]

[†]NTT Cyber Solutions Laboratories, NTT Corporation, 1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa, 239-0847 Japan
{kabutoya.yutaka,shiohara.hisako,fujimura.ko}@lab.ntt.co.jp
[‡]NTT Communication Science Laboratories, NTT Corporation, 2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
iwata@cslab.kecl.ntt.co.jp

Abstract

We propose a new method of recommending questions to answerers so as to suit the answerers' knowledge and interests in User-Interactive Question Answering (QA) communities. A question recommender can help answerers select the questions that interest them. This increases the number of answers, which will activate QA communities. An effective question recommender should satisfy the following three requirements: First, its accuracy should be higher than the existing category-based approach; more than 50% of answerers select the questions to answer according a fixed system of categories. Second, it should be able to recommend unanswered questions because more than 2,000 questions are posted every day. Third, it should be able to support even those people who have never answered a question previously, because more than 50% of users in current QA communities have never given any answer. To achieve an effective question recommender, we use question histories as well as the answer histories of each user by combining collaborative filtering schemes and content-base filtering schemes. Experiments on real log data sets of a famous Japanese QA community, Oshiete goo, show that our recommender satisfies the three requirements.

Introduction

User-interactive question answering (QA) communities, such as Oshiete goo[1] (OG) or Yahoo Answers, are gathering more and more attention. For example, more than 2,000 questions are posted every day in one of the most popular Japanese QA communities, OG. Other users solve posted questions by giving answers. Of course, the answers of each question can be reused by other users who have similar questions. In this way, QA communities enable us to share our knowledge. It is easy to extract latent needs from the text of questions and answers posted in QA communities, and thus provide effective advertisements that target the needs. This has made web service providers pay attention to the QA communities.

How satisfied are questioners with the answers provided by current QA communities? To answer this query we submitted a questionnaire to over 2,000 QA users about their use of QA communities. According to the results of the questionnaire, more than 90% of respondents were dissatisfied with the answers. The questionnaire also identified that answer quality is poor for about 50% of the dissatisfied questioners while answer quantity was perceived as by 25% of the dissatisfied questioners.

To overcome this dissatisfaction, we propose a method that can recommend particular questions to particular QA users. This question recommender reduces the time wasted by waiting for answerers to find suitable questions. If QA users could find the questions that suited them more efficiently, their motivation would be increased, and finally they would post more answers. Therefore, the number and quality of answers would both be enhanced.

Our questionnaire also showed that more than 80 % of respondents indicated that they would tackle questions that suited their knowledge and interests. In the QA system examined, each question is assigned to one category depending on its content (e.g., the question of "how to cook oatmeal" belongs to the "cooking" category). About 50% of the respondents used category information in selecting questions to tackle.

This approach is, however, rather impractical due to the enormous number of categories and the enormous number of questions posted in each category. Moreover, there are many categories that are similar to each other, which makes it difficult for questioners to select the most suitable category for their question. For example, a question about museums in New York should be posted in the "art" category or the "New York" category? If the question is posted in the former category, suitable answerers may miss the question if they restrict themselves to the latter category. This suggests the loss of latent answers.

To solve this problem, we propose a method that recommends to answerers those questions that are most relevant to the answerers' knowledge and interests; it is superior to the current category-based approach. Effective question recommenders must, however, meet several requirements other than high prediction accuracy. First, the effective question recommender is required to have the capability to recommend suitable questions to the users even who have not posted any answers previously: in OG, more than 60% of users have posted a question, but not an answer. Next, the recommender is also required to have the capability to rec-

[1]http://oshiete.goo.ne.jp/

Table 1: Notation

Variable	Description
Q	Question set
W	Word set
q	Question
$\boldsymbol{q} = \{q_w\}_{w=1}^{\|W\|}$	Keyword vector representation of q
U_q	User set answering q
u	User
Q_u^A	Question set questioned by u (u's question histories)
Q_u^R	Question set answered by u (u's answer histories)
$\boldsymbol{\theta}_u^A = \{\theta_{uw}^A\}_{w=1}^{\|W\|}$	Keyword vector representation of u's question histories
$\boldsymbol{\theta}_u^R = \{\theta_{uw}^R\}_{w=1}^{\|W\|}$	Keyword vector representation of u's answer histories
r	Whether a user answers a question
a	Whether a user posts a question
c_q	Category of q
f	Feature
F	Feature set

ommend even unanswered questions: in QA communities, more than 2,000 questions a day are posted, and such newly posted questions do not have any answers.

To achieve the question recommender that satisfies these requirements, we propose a method by combining *collaborative filtering* schemes and *content-base filtering* schemes. The collaborative filtering schemes are expected to yield high recommendation accuracy regardless of the questions content. On the other hand, the content-base filtering schemes are expected to yield the capability to recommend unanswered questions and to recommend to users who have never answered a question previously.

Proposed Method

We propose here a *question recommender* system (QR) as a function that supports answerers by finding questions that are relevant to their knowledge and interests. QR provides top-n questions ranked according to the relevancy between the answerer and each unanswered question.

Effective QR satisfies the following three requirements:

1. Higher relevancy than the existing category function.
2. Activating users who have never answered any question: whereas 253,072 users have given at least one answer, 327,669 users have never given anyanswer (but they have posted questions) in OG as of June, 2009.
3. Solution of unanswered questions.

We now describe the variables used in this section; they are described in Table 1. We exploit the *tf-idf* scheme to represent a question as a keyword vector (The document set for calculating idf values is Q).

Logistic Regression Model

Jin et al. (Jin, Zhou, and Mobasher 2005) showed through experiments that the accuracy of the logistic regression model (maximum entropy model) is high for general recommendations. Hence, we consider the logistic regression model to also be efficient in the question recommendation problem. When the logistic regression model is applied to our problem, the probability of user u answering question q can be modeled as follows:

$$P(r=1|u,q) = \frac{1}{1+\exp(-\sum_{f \in F} \lambda_f P_f(r=1|u,q) - \lambda_0)}, \tag{1}$$

where $\boldsymbol{\lambda} = \{\lambda_f\}_{f \in F}$ are unknown parameters.

We can obtain a global optimum solution for the unknown parameters $\boldsymbol{\lambda}$ by maximizing the following logarithmic posteriori against training data sets $\boldsymbol{X}$ using optimization techniques such as the quasi-Newton methods.

$$\begin{aligned} \log(P(\boldsymbol{\lambda}|\boldsymbol{X})) &\propto \log(P(\boldsymbol{\lambda})P(\boldsymbol{X}|\boldsymbol{\lambda})) \\ &\propto -\eta||\boldsymbol{\lambda}||^2 + \sum_u \sum_q r_{uq} \log(P(r=1|u,q)) \\ &\quad + \sum_u \sum_q (1-r_{uq}) \log(1-P(r=1|u,q)), \end{aligned} \tag{2}$$

where we assume that a prior $P(\boldsymbol{\lambda})$ follows a Gaussian distribution with hyperparameter η.

Features

We employ six features as follows:

1. **Probability to answer any questions in a category**
   Filtering questions against existing categories can be regarded as filtering questions by the probability of an answer existing in the category to which each question belongs.
   $$P_{\mathrm{CategoryA}}(r=1|u,q) = P(r=1|u,c_q). \tag{3}$$
2. **User-based collaborative filtering**
   This scheme is exploited by the recommender system called "Grouplens" proposed by (Resnick et al. 1994). This scheme has the advantage of yielding recommendations with high prediction accuracy regardless of question content information (this explains of why we adopt this scheme).
   $$P_{\mathrm{UserCF}}(r=1|u,q) = \frac{1}{|U_q|} \sum_{u' \in U_q} \cos(Q_u^R, Q_{u'}^R), \tag{4}$$
   where cos means cosine similarity.
3. **Item-based collaborative filtering**
   The recommendation scheme proposed by (Sarwar et al. 2001) offers reasonable computation cost and adequate prediction accuracy.
   $$P_{\mathrm{ItemCF}}(r=1|u,q) = \frac{1}{|Q_u^R|} \sum_{q' \in Q_u^R} \cos(U_q, U_{q'}). \tag{5}$$
   This scheme also has the advantage of not utilizing question content information.

4. **Content-base filtering using answer histories**
   This recommendation scheme was proposed by (Mooney and Roy 2000). We exploit the term weight (tf-idf value) of each word in the question as the content information. This scheme has, unlike collaborative filtering schemes, the ability to recommend (process) unanswered questions.

$$P_{\text{ContentA}}(r=1|u,q) = \cos(\boldsymbol{q}, \boldsymbol{\theta}_u^R). \tag{6}$$

5. **Content-base filtering using question histories**
   We consider that the knowledge and interests of a user can be estimated from the term weight of each word in the user's questions. The advantage of using question histories is that, unlike existing recommendation schemes, it makes it possible to recommend questions to users who have not answered any question up to now.

$$P_{\text{ContentQ}}(r=1|u,q) = \cos(\boldsymbol{q}, \boldsymbol{\theta}_u^A). \tag{7}$$

6. **Probability of posting a question in a category**
   We also exploit the category information of a user's question history as well as term weights.

$$P_{\text{CategoryQ}}(r=1|u,q) = P(a=1|u,c_q). \tag{8}$$

Experiments

A question that a user has actually answered can be regarded as a question that is relevant to the user's knowledge and interests; this is confirmed by the questionnaire results reported in Subsection 2.3. We evaluate our method described in Section 3 in terms of its ability to predict which question a user will answer.

Datasets

Our data is based on a snapshot of a Japanese famous QA community site, Oshiete goo, crawled in the period between May 22nd, 2009 and May 31st, 2009. We exploited all of the **24,272** questions in the snapshot above, all of the **53,354** answers, and all of the **5,559** answerers (users who gave least two answers in the period or gave least one question and one answer).

Evaluation Metric

We determined the accuracy of predicting which question each user will answer from the top-n accuracy. The procedure used to determine the top-n accuracy is as follows: First, the most recent question of those answered by user u is regarded as test data $\bar{q}_u$ for user u. Questions older than $\bar{q}_u$ are regarded as training data for u. Next, relevancy of u against $\bar{q}_u$ and each of the questions which u did not answer is calculated using recommendation methods. When $\bar{q}_u$ is included in the n most relevant questions to u, the correct questions are considered to have been recommended to u. The top-n accuracy is given by the fraction of users to which correct questions are recommended.

Methods Compared

The baselines are recommenders that use single features. We now describe our method and the baselines:

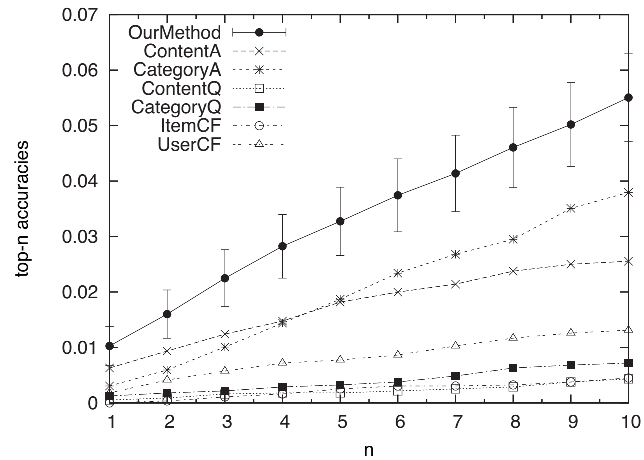

Figure 1: Top-n accuracies

- *OurMethod*. The hyperparameter used to estimate weight parameters in the logistic regression model is decided in the following manner: First, we set 10 candidates for hyperparameter η $\{10^{-5}, 10^{-4}, \ldots, 10^{4}\}$. Next, we calculate top-n accuracy where the test data for each user is taken to be the second-most recent question answered by each user under the condition that hyperparameter η is set to each candidate. Hyperparameter η is set to 10^{-2} since this maximizes the top-n accuracy.
- *CategoryA*. The recommender system uses the probability of each user being able to answer a question in each category. This method can be regarded as the existing category. Hence, comparing OurMethod against this baseline is very important.
- *UserCF*, *ItemCF*, *ContentA*, *ContentQ*, and *CategoryQ*: Each recommender system using a single feature.

Results

Top-n accuracy scores of each method are shown in Figure 1. The error bars of OurMethod line indicate the 99% confidence intervals of plots. First, we conducted a χ^2 test to determine if the difference between the top-n accuracy of OurMethod and that of CategoryA is significant. We now make null hypothesis H_0: the top-n accuracy scores of the two methods are not different from each other (and converse hypothesis H_1: their top-n accuracy scores are different from each other). The test rejects the null hypothesis with significance level of 0.01 for $\forall n \in [1, 10]$.

Consideration

The top-n accuracy of our method is higher than that of probability of answering in each category at the significance level of 0.01; therefore, OurMethod can pair answerers with more suitable questions than is possible withthe existing category approach.

The weights of schemes using answer histories are higher than those of schemes using question histories. The number

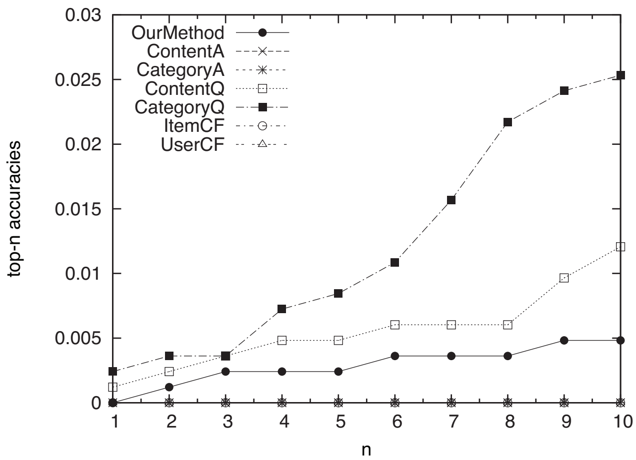

Figure 2: Top-n accuracy scores for users who have never answered

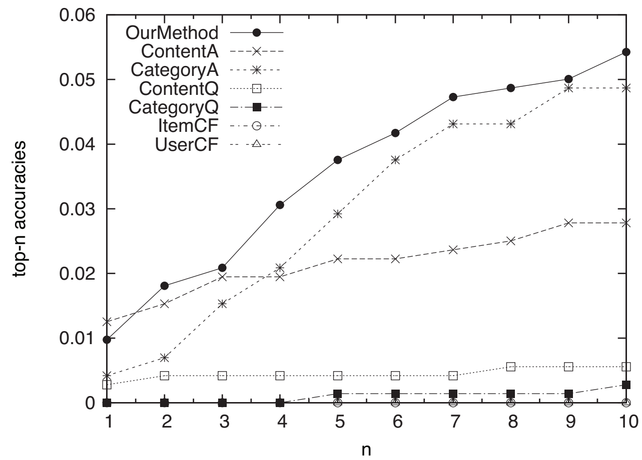

Figure 3: Top-3 accuracy scores versus number of questions answered

of a user's positive examples, used to estimate weight parameters, is equal to the number of questions that the user answered. Hence, the mote items the user's answer history contains, the more influence the user has on the estimation of weight parameters. This explains why answer histories are more efficient than question histories in recommending suitable questions to users who have answered many questions.

The lower weights of schemes that use question histories seems to suggest a fault with our method; suitable questions cannot be recommended to users who have only entered questions. That OurMethod can recommend questions to these users is validated in the analysis reported in Section 5.

Analysis

We consider that question recommender systems have several requirements other than high prediction accuracy. We now describe the requirements as follows:

- Questions should be recommended to users even if they have never answered a question before.
- Unanswered questions should be recommended.

We analyzed OurMethod in terms of these requirements above in this section.

Accuracy Scores for Questioners

First, we validate whether OurMethod is efficient regardless of the frequency of answering. Top-n accuracy scores for **829** users who have never answered but questioned are shown in Figure 2. The experimental result shows that our method is not efficient for such questioners. This weakness is caused by paucity of useful information contained within question histories.

Accuracy Scores for Unanswered Questions

Next, we confirmed whether OurMethod can efficiently handle unanswered questions. Top-n accuracy scores for **719** users whose test data is not unanswered by other users are shown in Figure 2. The result shows that OurMethod is efficient to recommend even unanswered questions. The recommenders that use content information in questions, such as CategoryA or ContentA, is trivially efficient against unanswered questions. On the contrary, recommenders based on collaborative filtering are not efficient since unanswered questions provide no useful information (this is called the "Cold-Start" problem).

References

Jin, X.; Zhou, Y.; and Mobasher, B. 2005. A maximum entropy web recommendation system: combining collaborative and content features. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 612–617. ACM.

Mooney, R., and Roy, L. 2000. Content-based book recommending using learning for text categorization. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, 195–204. ACM.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; and Riedl, J. 1994. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 175–186. ACM.

Sarwar, B.; Karypis, G.; Konstan, J.; and Reidl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, 285–295. ACM.