Probabilistic Question Recommendation for Question Answering Communities*

Mingcheng Qu[1], Guang Qiu[1], Xiaofei He[2], Cheng Zhang[3], Hao Wu[1], Jiajun Bu[1], and Chun Chen[1]
[1,2]College of Computer Science and Technology, Zhejiang University, China
[3]China Disabled Persons' Federation Information Center
[1]{qumingcheng, qiuguang, haowu, bjj, chenc}@zju.edu.cn,
[2]xiaofeihe@cad.zju.edu.cn, [3]zhangcheng@cdpf.org.cn

ABSTRACT

User-Interactive Question Answering (QA) communities such as Yahoo! Answers are growing in popularity. However, as these QA sites always have thousands of new questions posted daily, it is difficult for users to find the questions that are of interest to them. Consequently, this may delay the answering of the new questions. This gives rise to question recommendation techniques that help users locate interesting questions. In this paper, we adopt the Probabilistic Latent Semantic Analysis (PLSA) model for question recommendation and propose a novel metric to evaluate the performance of our approach. The experimental results show our recommendation approach is effective.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: information filtering

General Terms

Algorithms, Design, Experimentation

Keywords

Question Recommendation, Question Answering, PLSA

1. INTRODUCTION

Nowadays, the User-Interactive Question Answering (QA) community has become a popular medium for online information seeking and knowledge sharing. For example, Yahoo! Answers[1], one of the largest QA communities nowadays, has approximately 23 million resolved questions, which are posted and answered by users. In addition, there are also thousands of questions posted daily. However, with the exponential growth in data volume, it is becoming more and more time-consuming for users to find the questions that are of interest to them. As a result, the asker would have to wait for a long time before getting answers to his/her question.

To help users find interesting questions and expedite the answering of new questions, some question recommendation attempts are seen in QA communities like Yahoo! Answers, such as maintaining in user home pages a question list automatically generated based on features like posted time and ratings.

However, these systems are not typical recommender systems in essence in that they have not taken users' interest into account. In our work, We employ PLSA [3] to analyze a user's interest by investigating his previously asked questions and accordingly generate fine-grained question recommendation. Meanwhile, because traditional evaluation metrics cannot meet the special requirements of QA communities, we also propose a novel metric to evaluate the recommendation performance. Experimental results show the PLSA model works effectively for recommending questions.

2. PLSA FOR QUESTION RECOMMENDATION

Aiming to improve a QA community's efficiency, *question recommendation* is to recommend questions to users who are interested in, and capable of answering them. Therefore, the key to a question recommender is to capture users' interest.

In our work, we propose to analyze users' interest by investigating his previously asked questions. In a typical question answering cycle, users always answer questions by first identifying the topics in an implicit way. PLSA model [2], known for its ability of capturing underlying topics, suits our problem well. The latent variables in PLSA denote the topics of corresponding questions. Therefore, given a question collection, the distribution of users and their answered questions can be formulated as follows:

$$\Pr(u,q) = \sum_z \Pr(u|z)\Pr(q|z)\Pr(z) \qquad (1)$$

where $u \in u_1, u_2, ..., u_n$ are users, $q \in q_1, q_2, ..., q_m$ are questions and $z \in z_1, z_2, ..., z_k$ are k topic models, each capturing one topic u.

However, in a real QA community, each user can only answer a small percentage of the overall questions, which means that most observations (u,q) should be zero. In order to deal with sparsity, we use a user-word aspect model instead, where the co-occurrence data represent the event that users type words in a particular question:

$$\Pr(u,w) = \sum_z \Pr(u|z)\Pr(w|z)\Pr(z) \qquad (2)$$

*Supported by National Key Technology R&D Program of China (NO.2008BAH26B02)

[1] http://answers.yahoo.com

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

where $w \in w_1, w_2, ..., w_l$ are words which questions contain. Note that the PLSA model allows multiple topics per user, reflecting the fact that each user has lots of interest.

Then the log likelihood L of the question collection is

$$L = \sum_{u,w} c(u,w) \log \Pr(u,w) \tag{3}$$

where $c(u,w)$ is the sum of word w's count in all questions the user u answers.

Model parameters can be learned using Expectation Maximization (EM) to find a local maximum of the log likelihood of the question collection:

$$\begin{aligned}
\Pr(z|u,w) &= \frac{\Pr(u|z)\Pr(w|z)\Pr(z)}{\sum_{z'} \Pr(u|z')\Pr(w|z')\Pr(z')} && (4) \\
\Pr(u|z) &\propto \sum_{w} c(u,w)\Pr(z|u,w) && (5) \\
\Pr(w|z) &\propto \sum_{u} c(u,w)\Pr(z|u,w) && (6) \\
\Pr(z) &\propto \sum_{u,w} c(u,w)\Pr(z|u,w) && (7)
\end{aligned}$$

We then model recommending questions to users as the posterior probability $\Pr(u|q)$, that is, according to how likely it is that user u will access the corresponding question q. According to Bayesian law, we can compute $\Pr(u|q) \propto \Pr(u,q)$, which is calculated as the product of the probabilities of the words q contains, normalized by the question length:

$$\Pr(u,q) = \left(\prod_i \Pr(u, w_i) \right)^{1/|q|} \tag{8}$$

where w_i are words in the question q , and $|q|$ is the length of q. Consequently, a ranking list of users will be maintained for the question q according to the score. The recommendation can be conducted by recommending q to top-n users.

3. EXPERIMENTS AND RESULTS

To obtain the data sets for experiments, we crawl questions of three categories of the Yahoo! Answers: *Astronomy*, *Global Warming*, and *Philosophy*, and filter out all questions which have only one answer. Questions in each data sets are already labeled with the best answers. The data set statistics are listed in Table 1. For each category, a PLSA model is trained from 85% of the *question sets* (questions and their corresponding answers), and the left are used for testing. We empirically choose the number of latent variables $k = 100$.

In traditional recommender systems, we can use the *precision* to evaluate the performance. However, the precision metric fails to suit the QA context. Users in a QA community can only access a small portion of questions of all. While questions one accessed are those he/she is interested in, there is no guarantee that questions he/she has not accessed are those he/she does not like.

Here we propose a new metric for the evaluation of question recommendation. For each question in testing data, we only recommend it to the users who actually answered it instead of all possible users in the whole data sets. Then the accuracy for this question is defined according to the rank of the user who provides the best answerer. Since the choice of the *best answer* subjects to *asker*'s personal viewpoint, one may question whether the *best answer* is objectively the best of all, or just the *asker*'s prejudice. Adamic et al. [1] check questions from different categories in Yahoo! Answers, and draw the conclusion that answers selected as *best answers* are mostly indeed best answers for the questions. Therefore, in this paper we use the best answerer's rank as the ground truth of our evaluation metric:

$$accuracy = \frac{|R| - R_B - 1}{|R| - 1} \tag{9}$$

where $|R|$ is the length of recommending list, which is equally the number of answers in this *question set*, and R_B is the rank of the best answerer.

Table 1: Yahoo! Answers data set.

Category	Questions	Answers	Users
Astronomy	8,920	49,297	16,391
Global Warming	8,330	82,788	22,015
Philosophy	9,477	84,953	22,822

Table 2: Comparison of recommending methods.

Category	Cosine	PLSA
Astronomy	0.621	**0.648**
Global Warming	0.627	**0.674**
Philosophy	0.634	**0.709**

As there is no previous work done on recommending questions to users according to their interest in QA communities, for comparison we implement *Cosine Similarity* between user and question vectors, with tf.idf weights:

$$s(u,q) = \frac{\sum_w tf.idf(u,w) tf.idf(q,w)}{\sqrt{\sum_w tf.idf(u,w)^2}\sqrt{\sum_w tf.idf(q,w)^2}} \tag{10}$$

where $tf.idf(q,w)$ is the word w's tf.idf weight in q, and $tf.idf(u,w)$ is the sum of w's tf.idf weights in questions that u posts/answers.

Table 2 shows the experimental results. We observe that our PLSA model outperforms the cosine similarity measure in all the three data sets. It shows PLSA can capture users' interest and recommend questions effectively.

4. CONCLUSION

In this paper, we introduce the novel problem of question recommendation in Question Answering communities. We adopt the PLSA model to tackle this novel problem. We also propose a novel evaluation metric to measure the performance. The results show PLSA model can improve the quality of recommending. In conclusion, our study opens a promising direction to question recommendation.

5. REFERENCES

[1] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *WWW '08*.

[2] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99*.

[3] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *UAI '01*.