User-based Question Recommendation for Question Answering System

Gang Liu and Tianyong Hao

***Abstract*—An approach to automatically recommending question based on user-word model is proposed. We first employ language modeling approach to map the relationship between a user and a question into the relationship between the user and words in the question. A use-word model is then designed to reveal and quantify the affinity relationship between users and words in the corpus. In the recommendation model, a new question is assigned to users based on the evaluation of question-user relationship. The user who has the strongest relationship with the question is recommended to answer the question. We also implement an incremental update model which can dynamically maintain the user-word model. 216,563 questions (spreading into 30 categories) from Yahoo! Answers are collected as dataset and preliminary experiments show our approach achieves question recommendation accuracy by 85.2%, which exceeds baseline methods.**

***Index Terms*—Question recommendation, question answering, collaborative filtering**

I. Introduction

The past years has seen that research into the use of question answering to provide more precise answers to users' questions emerges increasingly popular. As a consequence, more and more user interactive question answering systems have been launched in recent years, including Yahoo! Answer [1], Ask Jeeves [2], and Baidu Zhidao [3]. These question answering systems provide opportunities for users to post their questions as well as to answer others' questions. However, due to the complexity of the human languages, most current question answering systems have difficulty in effectively analyzing users' free text questions. As a result, how to effectively and efficiently produce recommendation emerges as an important issue to solve such problems. To response to the problems, many researchers have engaged themselves into this field. Collaborative filtering (CF) [4] is an approach to filtering patterns and information by collaborating users' feedback, history data, etc. It is widely used in the recommendation systems. Hu et al. [5] proposed a balanced question recommendation approach to recommend a new question to qualified users for user-interactive question and answering system. In their method, a user modeling method is used to evaluate the interestingness and professionality of every user thus to select qualified users for a new question. Qu et al. [6] adopted the Probabilistic Latent Semantic Analysis (PLSA) model for question recommendation. They studied users' historically asked questions to evaluate users' interests. PLSA model is used to capture the underlying topics which users cared about. They discovered that user interest and question characteristics always had closed relationship with a few latent semantics. Those latent semantics could represent the topics to which users pay more attention. Wu et al. [7] proposed an Incremental Probabilistic Latent Semantic Analysis for automatic question recommendation. They designed an incremental algorithm which considered users' long-term and short-term interests. They defined long-term interest as all the questions that users already asked, while short-term interest as the new questions the users asked lately. They employed a Bayesian method proposed by Chou and Chen [8] to solve the problem of online event detection. When a new question came, they would modify the relationship from the latent topics to the question and the latent topics to words of that question accordingly.

In fact, with the accumulation of a huge number of questions and answers, in some user-interactive question answering systems, users' questions cannot be answered in time. It is also time-consuming for users to find their interesting questions online. Thus, a well designed mechanism which can provide a directing recommendation service is required such that it helps find suitable users who are interested in and capable of answering questions recommended.

In this paper, we propose a new approach to solve the problem of question recommendation based on a user-word model analysis, which can help users find interesting questions and expedite the process of answering new questions. The user-word model is designed to reveal the affinity relationship between users and words in the question corpus and quantify these relationships. Affinity relationship between a user and a word is used to depict the user's preference for the word. The more the words occur in a question which the user has strong affinity relationship with, the more attention on the question the user will pay. We first trains the user-word model for every user involved by employing a supervised term weighting method to weight every word in the whole question corpus. When a new question comes, the approach calculates the relationship between users and words in the question combining with the user-word model and recommends the question to the user who has the strongest affinity relationship with the given question. We also implement an incremental update model which can dynamically maintain the user-word model when the questions are answered by users. It can help the approach to avoid re-training on the whole corpus when the user-word model is changed by users answering the question. Through such approach, our question answering approach can estimate the interests and professional areas of each user and

Manuscript received March 19, 2012; revised April 28, 2012.

G. Liu and T. Hao are with the Department of Chinese, Translation and Linguistics, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

automatically recommend the new question to eligible users.

II. The User-based Question Recommendation Approach

The work flow of our question recommendation approach is shown in Fig. 1, which consists of four main models: Question Recommendation Model, User-Word Model, Training Model, and Incremental Update Model. The question recommendation model is responsible for assigning a new question to qualified users to answer. The user-word model is designed to reveal the affinity relationship between users and words in the question corpus and quantify these relationships. The function of training model is responsible for training the user-word model from the whole question corpus. It evaluates the user-word model for every involved user by using supervised term weighting method to weight each word in the whole question corpus. The incremental update model is the operation to incrementally update the user-word model when questions are answered by users. This model avoids re-training on the whole corpus when the user-word model is changed by users answering the question. It thus adapts to online recommendation services for its fast processing ability.

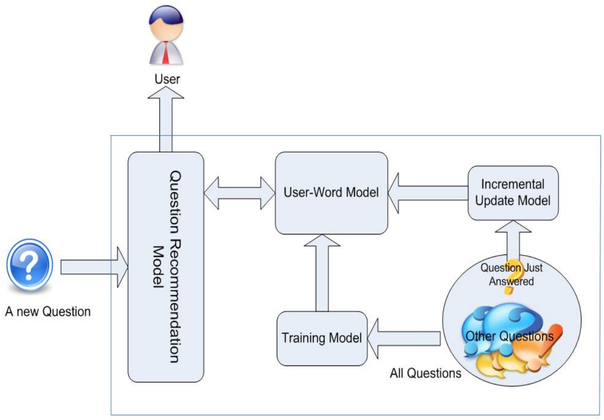

Fig. 1. The work flow of our user-based question recommendation mechanism.

In this approach, we consider each user as a different category.Thus the problem of recommending a new question to a qualified user is changed into the classification of the question to a suitable user (the category). In a real question answering community, each user would only answer a small percentage of the whole questions. For a question, the part capturing the user's interest would be some key words. The more the words occur in a question which the user has strong affinity relationship with, the more attention to the question the user will pay and the higher possibility our approach would recommend the question to that user. Hence we use a user-word model to reveal such relationship between users and words of the questions, and then recommend a question to a user to his/her interest.

A. Language Modeling Approach

The language modeling approach [8] has been widely applied in many applications, which ranks documents by the probability of generating the query terms in their language models. This model is used to assign a likelihood to a user's query $q=(q_1, q_2, \ldots, q_m)$. If a document d prior $p(d)$ is specified, the posteriori probability of a document is computed with the following equation.

$$p(d \mid q) \propto p(q \mid d)p(d) \tag{1}$$

In our work, we employ language modeling approach to calculate the relationship between users and query (i.e. question). We replace the document d with the user u and define the query q as a new question needed to be recommended. Hence we obtain equation (2) as follows.

$$p(u \mid q) \propto p(q \mid u)p(u) \tag{2}$$

Suppose we have a question collection C, we model the relationship between the user u and words w in the collection as follows.

$$p_\lambda(w \mid u) = \lambda \times p(w \mid u) + (1-\lambda) \times p(w \mid C) \tag{3}$$

where λ is a balance parameter. We employ the estimate translation models [10] for mapping a user term to the question terms. For a new question q which contains m words as $w=<w_1, w_2, \ldots w_m>$, using the translation models, we can obtain the value of $p(q|u)$ as follows.

$$p(q \mid u) = \sum_{i=1}^{m} t(q \mid w)p(w \mid u) \tag{4}$$

Hence, given a question q, we can calculate $p(u|q)$ based on deduction of equation (2), (3), and (4). The higher value of $p(u|q)$, the more qualified the user u to be suitable for answering the question q.

B. Term Weighting Method

In a question, different words have different importance to the question. For different users, even the same word would cause different interest to them. Thus the tradition *TF-IDF* method may not be satisfied with this requirement to measure the weight of a word. To differentiate users' interests to the same word, we need a method which could assign appropriate weight to the words. Hence we apply term weighting method [11] to measure the weight of the word in the corpus for different users as $p(w|u)$.

$$p(w \mid u) = \frac{(a \times d + b \times c)^2}{(a+c)\times(b+d)\times(a+b)\times(c+d)} \tag{5}$$

Algorithm 1: Our incremental update algorithm

Input: a question q is answered by a user u
Output: for all the words in q, output updates $P(w|u)$ for the user u

```
add question q and update the question list
process the question q: remove the stop words, stem the remaining words w into their root form
if user u is new then
    create user u,
    for all the word w do
        if word w is new then
            add w into words corpus and initialize its value;
        end if
        initialize the value for count of w given the user u;
        compute the probability of P(w|u) as equation 5 shows;
    end for
else /* Is not new user */
    for all the word w do
        if word w is new then
            add w into words corpus and initialize its value;
        end if
        update the value for count of w given the user u;
        compute the probability of P(w|u) as equation 5 shows;
    end for
end if
```

Fig. 2. The incremental update algorithm.

where, a is the number of questions answered by the given user who use this word; b is the number of questions answered by the user who do not use this word; c is the number of questions answered by other users who use this word; d is the number of questions answered by other users who do not use this word.

C. Incremental Update Method

When a user posts a new question or answers an existing question q, the degree of the given user's preference for the word $p(w|u)$ is updated accordingly. The description of the incremental update algorithm is illustrated in detail, as Fig. 2 shows.

III. Experiments and Evaluation

We collect questions from Yahoo! Answers as our dataset, which contains 216,563 questions spreading into 30 categories. The number of questions in each category varies from 3,659 to 9,500. Each question answered by at least 3 users. After the removal of stop words, each word is stemmed into its root form. The questions which contains too less words are filtered, i.e. "what time?" since such questions are meaningless for question recommendation.

A. Performance Metric

In order to evaluate the result of our question recommendation, we use 108,282 questions, half of the datasets, as the training dataset to train the user-word model. After that, we choose the remaining question as testing dataset to evaluate the performance of question recommendation. All questions in the testing dataset are already answered by the users.

The evaluation is based on correct recommendations over all recommendations, in which a correct recommendation is defined as: a given question is recommended to the user who has already answered the question before. We use the accuracy rate to quantify its performance, which is defined as follows:

$$Accuracy\ Rate = \frac{N_C}{N} \qquad (6)$$

where, N_C is the number of the questions which are correctly recommended; N is the total number of the questions we tested in the experiments. Our method's accuracy rate of question recommendation is 85.2%.

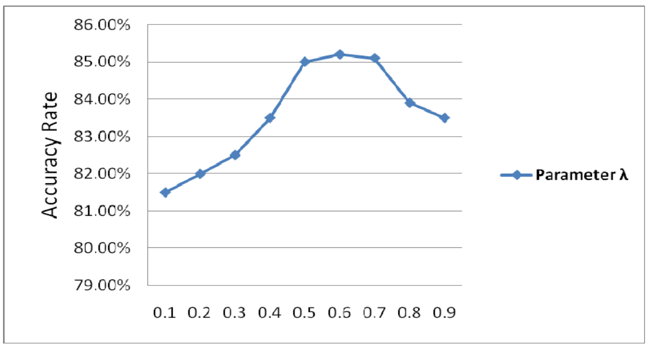

Fig. 3. Accuracy rate are evaluated with different λ.

B. Empirical Parameters Training

In this experiment, we tune the parameter λ, required in equation (3), to find optimal values so as to achieve maximum average accuracy rate in question recommendation. 108,281 questions in the testing dataset are applied using different values of λ from 0.1 to 0.9.

As the Fig. 3 shows, as λ is increased, the recommendation accuracy increases accordingly. When λ equals to 0.6, the accuracy rate obtains its maximized value. The finding suggests that considering the users' preference for words help improve the recommendation accuracy. The higher the value λ is, the more attention the approach pays to user's preference for the words. However, excessively depending on the users' reference for words would not help enhance the performance of the question recommendation. On the contrary, the recommendation accuracy is dramatically decreased as we can see from the figure when λ is set at 0.8 and 0.9. We conclude that our method provides the best performance with λ= 0.6, where the accuracy rate is 85.2%.

C. Comparison with Balanced Question Recommendation

We also compare our method with a balanced question recommendation mechanism for user-interactive QA system proposed by [5] in terms of performance. Hu's method first calculated the interestingness and professionality of each user and then selected the most qualified user for the given question.

Experimental results, as shown in Table 1, display that our method has a higher accuracy even on a much larger dataset. The accuracy of our method in question recommendation is improved by 4.4% compared with that of Hu's method. Hence our method has more advantages over Hu's method in overall performance.

TABLE I: Performance Comparison of the Two Methods

Recommendation methods	# of training questions	# of testing questions	Accuracy rate
Hu's method	50 questions	650 questions	80.8%
Our method	108,282 questions	108,181 questions	85.2%

IV. Conclusion and Future work

In this paper, we presented a fast user-based question recommendation for question answering system. We designed a user-word model to reveal the affinity relationship between users and words of the questions, and then recommended a question to a user to his/her interest. 216,563 questions were used in the experiment and the accuracy rate of our method's recommendation reached 85.2%. It can satisfy the need of online recommendation for the question answering system. In the future, we will intend to investigate and evaluate more accurate and compatible method to evaluate the user model analysis in the question recommendation thus to further improve the overall performance of the proposed method. We will also explore more useful factors such as the users' reputation, degree of activeness, and the quality of users' answers to improve the accuracy of the question recommendation approach.

References

[1] Yahoo! Answer. Retrieved on March, 2012, from http://answers.yahoo.com/.

[2] Ask Jeeves. Retrieved on March, 2012, from, http://uk.ask.com/.

[3] Baidu Zhidao. Retrieved on March, 2012, from http://zhidao.baidu.com/.

[4] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use, in *Proc. of the SIGCHI conference on Human factors in computing systems (CHI'95)*. pp. 194-201,1995.

[5] D. Hu, S. Gu, S. Wang, Liu W., and E. Chen. Question recommendation for user-interactive question answering systems. *Proc. of the Second International Conference on Ubiquitous Information Management and Communication (ICUIMC'08)*. pp. 39-44,2008.

[6] 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 the 18th international conference on World wide web (WWW'09)*. pp. 1229-1230,2009,

[7] H. Wu, Y. Wang, and X. Cheng, "Incremental probabilistic latent semantic analysis for automatic question recommendation," in *Proc. of the 2008 ACM Conference on Recommender Systems (RecSys'08)*. pp. 99-106, 2008.

[8] T. Chou and M. Chen, "Using incremental PLSI for threshold-resilient online event analysis," *IEEE Transactions on Knowledge and Data Engineering*. pp.289-299, 2008.

[9] J. Lafferty and C. Zhai, "Document language models, query models, and risk minimization for information retrieval," in *Proc. the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'01)*. pp. 111-119, 2001.

[10] D. Miller, T. Leek, and R. Schwartz, "A hidden Markov model information retrieval system," in *Proc. of 1999 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*. pp. 214-221,1999.

[11] M. Lan, C. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE transactions on pattern analysis and machine intelligence*, pp. 721-735, 2009.