Learning to Recommend Questions Based on Public Interest [*]

Jun Wang
State Key Laboratory of Software Development Environment
Beihang University, Beijing 100191
junwangbuaa@gmail.com

Xia Hu
Department of Computer Science and Engineering
Arizona State University, Tempe 85281
xiahu@asu.edu

Zhoujun Li
State Key Laboratory of Software Development Environment
Beihang University,Beijing 100191
lizj@buaa.edu.cn

Wenhan Chao
Beihang University
Beijing 100191
chanwenhan@buaa.edu.cn

Biyun Hu
Beihang University
Beijing 100191
byhu8210@gmail.com

ABSTRACT

This paper is concerned with the problem of question recommendation in the setting of Community Question Answering (CQA). Given a question as query, our goal is to rank all of the retrieved questions according to their likelihood of being good recommendations for the query. In this paper, we propose a notion of public interest, and show how public interest can boost the performance of question recommendation. In particular, to model public interest in question recommendation, we build a language model to combine relevance score to the query and popularity score regarding question popularity. Experimental results on Yahoo!Answers dataset demonstrate the performance of question recommendation can be greatly improved with considering the public interest.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Public Interest, Question Recommendation, CQA

[*]This work is supported by State Key Laboratory of Software Development Environment(SKLSDE-2011ZX-03).

1. INTRODUCTION

Community Question Answering (CQA), in which users answer questions posted by others, is gaining a large number of audiences in recent years, and has accumulated millions of questions and answers over time. These CQA portals provide an alternative channel for obtaining information on the web: rather than browsing results of search engines, users present detailed information needs and get direct response authored by humans [1].

With the exponential growth in data volume, the problem in CQA portals is how to answer users' questions efficiently. It was observed that in Yahoo!Answers[1] only 17.6% of questions received satisfied answers within 48 hours and nearly 20% of unresolved questions received no response [4], which showed that the popular CQA portal performs poorly in solving users' questions directly. This problem gives rise to question recommendation techniques that help users locate interesting questions. To date, most previous work [4] [6] have been done to solve this problem by analyzing users' information, such as the users' interests, whether she is an expert or is available to provide an answer. Unfortunately, the fact is that users usually are reluctant to provide their personal information when they browse or search question. Thus, it is relatively hard to predict whether a question is interesting to a specific user when the profile information of the user is not available [6].

In the setting of CQA, many users share the same or similar interests. Usually, "the wisdom of the crowds" for information needs are expressed by some frequently asked questions. If these popular questions can be identified and be used for recommending, we believe it can save much time for most users either in crafting successful questions or obtaining satisfied answers. Based on this observation, in this paper, we attempt to employ public interest to boost question recommendation performance. We try to model the question popularity to measure the public interest. Two perspectives are considered in the popularity measurement — possibility of question repeated by other people and users' response to this question. To the best of our knowledge, this is the first attempt of learning to recommend questions from

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

this aspect. In the question recommendation framework, we first use the question popularity to model the public interest. Then we build a language model for question recommendation, which combines relevance score to the query and popularity score regarding question popularity.

2. QUESTION RECOMMENDATION BASED ON PUBLIC INTEREST

An ideal question that should be recommended to a query should have 1) a high relevance score to the query; 2) a high question popularity score among related questions. As the question popularity score is independent to the query, it can be calculated offline. In this section, we first present our method on the question popularity measurement. Then we build a language model for question recommendation, which naturally combines the popularity score and relevance score.

2.1 Question Popularity

We drive the popularity of a question from two observations — the probability that the question would be repeated by other people and the people's response to this question. The method is based on two assumptions:

Assumption 1: The higher possibility would be a question repeated by other people, the more popularity the question is.

Assumption 2: The more responses does a question receive, the more popularity the question is.

2.1.1 Question Popularity Based on Question Repeated Possibility

If a topic is very interesting to people, there can be many lexically similar questions related to the topic in the question collection. Among those similar questions, the most central questions with the highest repeated possibility can be regarded as the most popular ones regarding the topic. To measure the question repeated possibility, we use the LexRank Method [3]. The LexRank estimates the centrality(repeated possibility) of a question in a manner similar to the PageRank [2].

First, we construct an undirected graph of questions based on similarities between each two questions. As the question in CQA is usually a short sentence, it may occur sparseness problem when using cosine method to measure the similarity between questions. Since the answers in CQA systems are manually crafted by users, they are usually more comprehensive which can provide more context information than questions. Intuitively, two questions should be semantically similar if their corresponding answers are similar, even though there is little word overlap between the two questions. Based on this observation, we use the answers similarity to improve the question similarity measurement.

Let i and j be two questions, $ans(i)$ and $ans(j)$ be the answer of i and j respectively. The similarity between i and j is defined as:

$$S(i,j) = \frac{sim(i,j) + sim(ans(i), ans(j))}{2} \tag{1}$$

The function $sim(\cdot)$ determines the lexically similarity between two sentences. Clearly, there are many possible selections for the similarity method, we choose cosine here and leave other methods for future work.

An undirected graph is constructed based on the similarities between questions. In the graph, each node represents a question and two nodes are connected if the similarity between them exceeds a certain threshold value. In our work, we set the threshold to be 0.5. If the normalized similarity exceed 0.5, then there is an edge between the two questions.

Second, for a question u, the repeated possibility $p(u)$ is calculated with the following weighting scheme:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} \frac{S(u,v)}{\sum_{z \in adj[v]} S(z,v)} p(v) \tag{2}$$

where N is the total number of nodes in the graph, d is a "damping factor", which is typically chosen in the interval [0.1, 0.2] [2], and $adj[u]$ is the set of questions that adjacent to u. $S(u,v)$ is defined in eq.1. Erkan and Radev presented an algorithm for the eq.2. For more details, please see [3].

2.1.2 Question Popularity Based on Response Information

According to Assumption 2, the more responses does a question receive, the more popularity the question is. In this paper, we explore the answers statistic features to estimate the response information and leave other features such as user ratings or comments for future work. For a question q, the impact of response information $w(q)$ on q's popularity is defined as :

$$w(q) = f(q, an_q) / \sum_{p \in C} f(p, an_p) \tag{3}$$

where C denotes the collection of questions, an_q is the number of answers of q, and $f(p, an_p)$ is defined as:

$$f(q, an_q) = \begin{cases} an_q & \text{if } an_q < cutN \\ cutN & \text{otherwise} \end{cases} \tag{4}$$

where $cutN$ is an integer which is supposed to be the largest number of answers of a question in a specific collection. According to the statistic information of the question collection in our dataset, we set $cutN$ to be 30.

Considering both impacts of question repeated possibility and responses information, the final function of question popularity is defined as:

$$Pop(q) = d \cdot w(q) + (1-d) \sum_{q' \in adj[q]} \frac{S(q,q')}{\sum_{u \in adj[q']} S(u,q')} Pop(q') \tag{5}$$

where $w(q)$ is defined in eq.3

2.2 Question Recommendation

Given a question as a query q', the recommendation model is defined as the probabilistic of generating query q' from the question language model q as follows:

$$P(q|q') = \frac{P(q'|q)P(q)}{P(q')} \tag{6}$$

As the likelihood $P(q')$ of q' does not affect the ranking of questions so it can be ignored by the rank preserving the principle.

The generation probability $P(q'|q)$ can be decomposed into a unigram model:

In eq.6, $P(q)$ is the prior probability of question q, reflecting a static rank of the question, independent on query q'. As the question popularity $Pop(q)$ defined in section 2.1 is the likelihood of a question and independent on a specific query, we can use the question popularity $Pop(q)$ as the prior probability of question q. So the probability of question q should be recommended to query q' is defined as:

$$R(q|q') \propto \alpha logPop(q) + \sum_{q \in q'} logP(t|q') \tag{7}$$

where α is a constant factor to indicate the importance of question popularity, which will be discussed in experiments. The unigram probability P(t|q') is defined as eq.8:

$$P(t|q') = \lambda P_{mle}(t|q') + (1-\lambda)P_{mle}(t|C) \tag{8}$$

where

$$P_{mle}(t|q') = \frac{\#(t,q')}{|q'|}, P_{mle}(t|C) = \frac{\#(t,C)}{|C|}$$

and $\#(t,q')$ denotes the time that term t appeared in q', $|q'|$ denotes the length of q', C denotes the question collection, λ is a smoothing parameter and set to 0.2 in our experiments, the same as [7][5].

3. EXPERIMENTS

3.1 Data Set and Experiment Setup

We collected 17,680 questions in "diet & fitness" category from Yahoo!Answers as the experiment dataset. We randomly select 50 questions as query questions and use different methods to recommend questions to the query. Top 10 results of each query were judged by human. First, we employ different method to recommend questions to each query. For each method, we collect the top 10 results of each query. Then we combined different results together by query. Then an assessor is asked to label it with 'relevant' or 'irrelevant'. If a returned result is considered as a good recommendation for the queried question, the assessor will label it 'relevant'; otherwise, the assessor will label it 'irrelevant'. A good recommendation of a query means that the recommended question should not only be relevant to the query but also be representative among the related questions.

We want to investigate the effectiveness of question popularity to the performance of question recommendation. Thus we compare the performance of the following methods.

- **Rec_NoP**: Each question has equal question popularity score.
- **Rec_QP1**: Question popularity is estimated by question repeated possibility based on Assumption 1 and question similarity is measured by cosine method on questions.
- **Rec_QR2**: Question popularity is estimated by question repeated possibility based on Assumption 1 and question similarity is measured by improved question similarity measurement (eq.1). The question popularity is calculated with eq.2;
- **Rec_AP**: Question popularity is estimated based on Assumption 2 with eq.3. The recommendation function becomes:

  $$R(q|q') \propto \alpha logw(q) + \sum_{t \in q'} logP(t|q')$$

- **Rec_QAP**: Question popularity is estimated based on both Assumption 1 and Assumption 2 with eq.5. Eq.7 is used as the recommendation function.

We use MAP (Mean Average of Precision), MRR (Mean Reciprocal Rank), and Precision@10 as metric to evaluate the performance of each method.

MAP calculates the mean of average precisions over a set of queries. MAP is calculated as:

$$MAP = \frac{1}{|Q_r|} \sum_{q \in Q_r} \frac{\sum_{r=1}^{n} P(r)rel(r)}{|R_q|} \tag{9}$$

where Q_r is a set of test queries, R_q is the set of relevant questions for q, r is the rank, n is the number of returned results, here $n = 10$, $rel(\cdot)$ is a binary function on the relevance of a given function, and $P(\cdot)$ is precision at a given cut-off rank.

MRR is calculated as :

$$MRR = \frac{1}{|Q_r|} \sum_{q \in Q_r} \frac{1}{r_q} \tag{10}$$

where Q_r is a set of test queries, r_q is the rank of the first relevant question for q. Precison@n is the fraction of the top n questions recommend that are relevant, it is calculated as:

$$Precision@n = \frac{\sum_{i=1}^{n} rel(i)}{n} \tag{11}$$

where $rel(\cdot)$ is a binary function on the relevance of a given rank, and here $n = 10$.

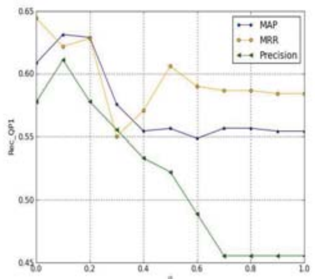
3.2 Impact of Question Popularity

In order to evaluate the effect of question popularity, controlled by the value of λ , for question recommendation, we tested different settings of λ with different question popularity estimation methods. Figure 1 gives the performance of Rec_QP1, Rec_QP2, Rec_AP and Rec_QAP with different settings of. From Figure 1 (a) and (b) we find that when is setting around 0.1, Rec_QP1 performs best, around 0.3, Rec_QP2 performs best, around 0.4, Rec_AP and Rec_QAP performs best.

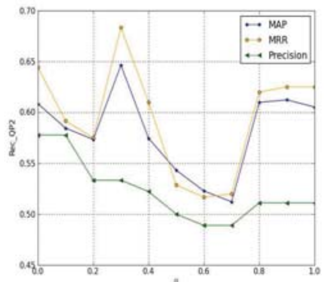
3.3 Experiment Results

Table 1 reports each method's performance measured by MAP, MRR and Precison@10. Each row reports the results of a method and the improvement compared with other methods.

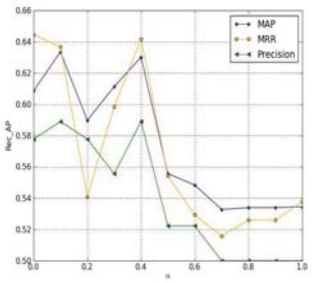
- **Rec_QP1** vs.**Rec_QP2**
  Both of the two methods are based on Assumption 1 when calculating the question popularity. Rec_Q1 employs Song et al[4] method to measure the similarity between questions while Rec_QP2 employs our improvement question similarity measurement. The results in table 1 show that Rec_QP2 out performs Rec_QP1 in terms of MAP and MRR but performs poorly in Precison@10. This result indicates that our proposed method on question similarity measurement can partially improve recommendation performance.
- **Rec_AP** vs.**Rec_NoP**
  Rec_AP is based on Assumption 2 when calculating the question popularity. The results in table 1 show that Rec_AP performs little better than Rec_NoP in MAP but worse in MRR and Precision@10. This result indicates that just using the number of answers of a question to estimate question popularity improve question recommendation performance hardly. Thus,



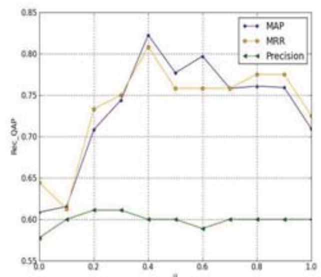
(a) Rec_QP1



(b) Rec_QP2



(c) Rec_AP



(d) Rec_QAP

Figure 1: The effect of question popularity with different methods. (a) Rec_QP1: question popularity is estimated by question repeated possibility and question similarity is measured by cosine method on questions (b) Rec_QP2: Question popularity is estimated from question repeated possibility with improved question similarity measurement. (c) Rec_AP: question popularity is estimated from the responses to questions (d) Rec_QAP: the combination of method (b) and (c)

Table 1: Question recommendation performance on Yahoo! Answers dataset

Methods	*MAP*	*MRR*	*Precision@10*
Rec_NoP	0.608571	0.6444444	0.5777778
Rec_QP1	0.631433	0.6217857	0.6111111
imp.Rec_NoP	3.76%	-3.52%	5.77%
Rec_QP2	0.646389	0.6833333	0.5333333
imp.Rec_NoP	6.21%	6.03%	-7.69%
imp.Rec_QP1	2.37%	9.90%	-12.73%
Rec_AP	0.63345	0.6366667	0.5888889
imp.Rec_NoP	4.09%	-1.21%	1.92%
imp.Rec_QP1	0.32%	-2.39%	-3.64%
Rec_QAP	0.822738	0.808333	0.6
imp.Rec_Nop	35.19%	25.43%	3.85%
imp.Rec_QP1	30.30%	30.00%	-1.82%
imp.Rec_QP2	27.28%	18.29%	12.50%
imp.Rec_AP	29.88%	26.96%	1.88%

in our experiments, we use the combination of question repeated possibility and number of answers information for further improvement of question popularity estimation.

- **Rec_QAP** vs.**Rec_NoP**
  Compared with the baseline method Rec_NoP, our method Rec_QAP consistently shows better performance in terms of all the evaluation measures. It can support our hypothesis that question popularity driven from both question repeated possibility and number of answers can improve question recommendation performance significantly.

4. CONCLUSIONS

In this paper, we propose to employ public interest to boost the performance of question recommendation. The question popularity is used to measure the public interest on a question. Two factors, named question repeated possibility and questions' responses information, are examined in assessing question popularity. Experimental results on Yahoo!Answers dataset demonstrate that leveraging public interest can significantly improve the performance of question recommendation. In addition, utilizing similarity between answers and questions can provide more accurate questions popularity estimation and thus give a better question recommendation performance. Furthermore, questions' responses information can also boost the performance of question recommendation. Experiments demonstrate that the combination of question repeated possibility and questions' responses information gives a better estimation on question popularity and greatly boosts the performance of question recommendation. However, our experiments have conducted with one domain. Additional experiments on larger and heterogeneous collections are essentially required for deep analysis of our proposed method. It will be our future work.

5. REFERENCES

[1] E. Agichtein, C. Castillo, and D. Donato. Finding high-quality content in social media. In *WSDM'09*, pages 183–193, 2009.
[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.
[3] G. Erkan and D. R. Radev. Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.
[4] B. Li and I. King. Routing questions to appropriate answerers in community question answering services. In *CIKM'10*, pages 1585–1588, 2010.
[5] A. S. M A Suryanto, Ee-Peng Lim and et al. Quality-aware collaborative question answering methods and evaluation. In *WSDM'09*, pages 142–152, 2009.
[6] K. Sun, Y. Cao, X. Song, and et al. Learning to recommend questions based on user ratings. In *CIKM'09*, pages 751–758, 2009.
[7] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR'01*, pages 334–342, 2001.