

# Finding Hard Questions by Knowledge Gap Analysis in Question Answer Communities

Ying-Liang Chen and Hung-Yu Kao\*

Department of Computer Science and Information Engineering,  
National Cheng Kung University, Tainan, Taiwan, R.O.C.  
hykao@mail.ncku.edu.tw

**Abstract.** The Community Question Answer (CQA) service is a typical forum of Web2.0 in sharing knowledge among people. There are thousands of questions have been posted and solved every day. Because of the above reasons and the variant users in CQA service, the question search and ranking are the most important researches in the CQA portal. In this paper, we address the problem of detecting the question being easy or hard by means of a probability model. In addition, we observed the phenomenon called knowledge gap that is related to the habit of users and use knowledge gap diagram to illustrate how much knowledge gap in different categories. In this task, we propose an approach called knowledge-gap-based difficulty rank (KG-DRank) algorithm that combines the user-user network and the architecture of the CQA service to solve this problem. The experimental results show our approach leads to a better performance than other baseline approaches and increases the F-measure by a factor ranging from 15% to 20%.

**Keywords:** CQA portal, difficulty, knowledge gap, link analysis, expert finding.

## 1 Introduction

Recently, the forum system of Web 2.0 is more and more popular and interesting. People can share or seek any information from any place in the world. One of the most popular and useful forum system is Community-based Question-Answering (CQA) portals. For example, the typical CQA portals such as Yahoo! Answers<sup>1</sup> in English, Naver<sup>2</sup> in Korean, and Baidu Knows<sup>3</sup> in Chinese, can be regarded as variations of online forums. In this paper, We choose Yahoo!Answers where approximately over one hundred million resolved questions in English for our research .

On account of the increasing progressively increasing questions and answers, user may not find the questions they want to know or discuss efficiently. Thus, most of

---

\* Corresponding author.

<sup>1</sup> <http://answers.yahoo.com>

<sup>2</sup> <http://www.naver.com>

<sup>3</sup> <http://zhidao.baidu.com>

works in CQA service are to improve their functionalities like question ranking, question search, and question recommendation. However, the output of the prior work do not consider the expertise (or authority) of users, i.e., for question ranking, the user may be an amateur in this area and the outputs of the search engine are so hard to understand for them. Moreover, an expert may want to search the harder questions but the top results are too easy to contain rich information for them. Thus, in this paper, *we concerned with how to rank questions by their difficulty levels.*

Moreover, the work of expert finding is highly associated with our task. The expert can be considered a user who is familiar with the particular topic or category, and the expert can solve most of questions in the particular category. In short, the difference between an easy question and a hard question is the ratio of non-experts and experts participating in this question respectively, and we can separate the task as the two parts: expert finding and question ranking. In this paper, the “expert” we want to find is different from the prior works, comparatively, we want to find the *answer-hard-question-frequently expert*. That is to say, we want to find the user that not only is an expert but also focuses on answering hard questions more than other users.

However, the fact in the real world is that the experts not only answer hard questions but also easy questions and it bring about the difficulty of determining a question is easy or hard. We observe a phenomenon called the *knowledge gap* existing in the CQA service. The knowledge gap is a phenomenon which associates with the expertise of users. There are two main principles on the knowledge gap. First, *the non-experts have no ability to answer the question that over their knowledge*. Second, *the experts have less interesting in the question that under the certain the degree of difficulty*. In this paper, we investigate the knowledge gap how to effect a CQA service and we propose a knowledge-gap-based difficulty rank (KG-DRank) algorithm calculating in probability to tackle the task that determine the question is easy or hard.

The rest of this paper is structured as follows. In Section 2, we briefly discuss related work. In Section 3, we define our task by the probability model and we present our approach called KG-DRank algorithm to solve this task. Next, experimental results are reported in Section 4. Section 5 concludes this work and our future work.

## 2 Related Work

There are several researches focusing on how to find expert by modeling and computing user-user graph such as [6,11,14]. McCallum [11] et al. utilize the user-user graph to find the experts for particular topics and then Zhang et al. [14] use the network-based ranking algorithms HITS [9] and PageRank [2] to identify users with high expertise. Their results show high correlation between link-based metrics and the answer quality. Next, Liu et al [10] use the features such as author’s activity, number of clicks for finding the best answers for a given question. Then [6,7] use HITS algorithm to find the authority of users in question/answering network that an asker is linked to an answerer if the answerer replies the question that asker has been asked. The experiments show that the obtained authority score is better than simply counting the number of answers an answerer has given. Although the results demonstrate that HITS algorithm is a good approach to find expert, but there are some potential factor like the habit of asking questions and replying answers may reduce the performance. Unlike [6] and [14], Zhou et al. [15] utilize the structural relations among users in the forum system and use content-based probability model to find the experts for the particular question.

Question ranking is to rank the results for purpose of browsing-time decreasing, the typical approach is using the information of Q&A content such as best ratings such as [3]. Jeon et al. [8] addressed the answer quality problem in a community QA portal and tried to estimate it, they used a set of non-textual features such as answer length, number of points received, etc. for determining answer quality. Agichtein et al. [1] expands on [8] by exploring a larger range of features including both structural, textual, and community features. Su et al. [13] analyzed the quality of each answers varies significantly. Bian et al. [3] proposed to solve collaborative QA by considering both answer quality and relevance and also used content-based quality answers without considering user expertise. Because of the answer quality problem on [3,8,13], Suryanoto et al. [12] propose a quality-aware framework that considers both answer relevance and answer quality derived from answer features and expertise of answerers.

### 3 Problem Definition and Our Approach

#### 3.1 Problem Definition

We state the problem of the degree of difficulty on questions  $q$  is by means of a probability model. Given a question  $q$ , the probability of the question  $q$  being hard is estimated as follows:

$$p(h|q) = \frac{p(q|h)p(h)}{p(q)} \quad (1)$$

where  $p(h)$  is the probability of a question being hard in the particular category and  $p(q)$  is the probability of a question generated by users, and the two variables are the same for all questions. Thus, our task is therefore to capture how much hard on the question  $q$  by  $p(q|h)$ .

#### 3.2 The KG-DRank Algorithm

The progress of our system only has iterations of the two steps: the expert finding and the difficulty degree detecting. The expert finding concludes the expertise model and reinforcement model, while the difficulty degree detecting concludes the difficulty degree model and question ranking model.

##### 3.2.1 Expertise Model

We cite the prior work [6] as our expertise model.

##### 3.2.2 Difficulty Degree Model

The question can be divided to the two parts: the asking and answering. Thus, the probability of a question  $q$  given the degree of difficulty being hard is obtained by taking the summation across the users of the two parts on the question:

$$p(q|h) = \alpha * p(c_q^a|h) + (1 - \alpha) * p(c_q^r|h) \quad (2)$$

Where  $p(c_q^a|h)$  represents how much hard in asking, and while  $p(c_q^r|h)$  represents how much hard in answering. In asking, we capture an estimate of  $p(c_q^a|h)$  by means

of three types of architecture of CQA service, and it means how much expertise asker has in asking. It can be expressed as:

$$p(c_q^a|h) = p(c_q^a|q) * p(q|a) * p_{ask}(a|\theta h) \tag{3}$$

Where  $p(c_q^a|q)$  is the probability of the content that asker  $a$  asks on the question  $q$  and  $p(q|a)$  is how much knowledge that the asker  $a$  contribute to the question  $q$  and  $p_{ask}(a|\theta h)$  is the probability of an asker-expert of the asker  $a$  given the difficulty degree model  $\theta h$ . The value of  $p_{ask}(a|\theta h)$  is initialized to the hub value that section 3.2.1 referred and we will discuss the change of the next value later in this section.

To compute an estimation of  $p(c_q^a|q)$ , we use the length of the content and normalize it by the square root such as the following:

$$p(c_q^u|q) = \frac{\sqrt{l_q^u}}{\sqrt{\sum_{i \in q} l_q^i}} \tag{4}$$

Where  $c_q^u$  is the content that user  $u$  has posted on the question  $q$  and  $l_q^u$  is the length of  $c_q^u$  and  $\sum_{i \in q} l_q^i$  is the summation of the length of all the posts on the question  $q$ . According to above estimation, the estimation of  $p(q|a)$  is defined by:

$$p(q|u) = \frac{\sqrt{l_q^u}}{\max_{q' \in u} \sqrt{l_q^{u'}}} \tag{5}$$

Where  $\max_{q' \in u} \sqrt{l_q^{u'}}$  is the maximum of the length that user  $u$  had asked before. The estimation of  $p(q|u)$  is similar to  $p(c_q^u|q)$ , but we use maximum instead of summation to compute  $p(q|u)$ . It is reasonable that the hard questions must be using more words to ask and the easy questions instead. For the same asker, the question which the asker uses more words to describe has stronger probability of being hard question than the other questions the same asker asked.

In answering, we also divided it into two parts: the best answerer and the other repliers, so how much difficulty given by the answerers can be obtained as follows:

$$p(c_q^r|h) = \beta * p(c_q^b|h) + (1 - \beta) * p(c_q^{r'}|h) \tag{6}$$

### 3.2.3 Reinforcement Model

The main idea of reinforcement model is that the more hard questions user participates in, the more expertise user obtains and the expertise can be divided into the asking and the answering. In this model, we define two types of association between difficulty and users. One of them is defined as :

$$p_{ask}(h|u) = \frac{|A_u \cap H|}{|A_u|} , \quad p_{ans}(h|u) = \frac{|R_u \cap H|}{|R_u|} \tag{7}$$

Where  $|A_u|$  is the number of questions that user has asked, while  $|R_u|$  is the number of questions that user has replied. And  $|A_u \cap H|$  represents the number of hard questions that user has asked, while  $|R_u \cap H|$  represents the number of hard questions that user has replied, furthermore, we use equation 9 to represent the notation *local difficulty* probability model. The other association between difficulty degree and user is that how

much ratio of hard questions user participates in his/her all participating questions, and it can be expressed as:

$$p_{ask}(u|h) = \frac{|A_u \cap H|}{|H|}, \quad p_{ans}(u|h) = \frac{|R_u \cap H|}{|H|} \quad (8)$$

Where  $|H|$  is the total number of hard questions in category C, and we called this equation as *global difficulty* probability module. Therefore, the global KG-DRank (**GKG-DRank**) of  $p_{ask}(u|\theta h)$  and  $p_{ans}(u|\theta h)$  is represented as:

$$p_{ask}(u|\theta h)^{k+1} = \lambda * p_{ask}(u|h) + (1 - \lambda) * p_{ask}(u|\theta h)^k \quad (9)$$

$$p_{ans}(u|\theta h)^{k+1} = \lambda * p_{ans}(u|h) + (1 - \lambda) * p_{ans}(u|\theta h)^k \quad (10)$$

Where  $p_{ask}(u|\theta h)^k$  and  $p_{ans}(u|\theta h)^k$  are the kth-iteration-number score of  $p_{ask}(u|\theta h)$  and  $p_{ans}(u|\theta h)$  which  $p_{ask}(u|\theta h)^0$  and  $p_{ans}(u|\theta h)^0$  are hub and authority that base HITS algorithm computes in the prior section, and  $\lambda \in [0,1]$ . The other way is local KG-DRank (**LKG-DRank**) is represented as :

$$p_{ask}(u|\theta h)^{k+1} \propto \lambda * p_{ask}(h|u) + (1 - \lambda) * p_{ask}(u|\theta h)^k \quad (11)$$

$$p_{ans}(u|\theta h)^{k+1} \propto \lambda * p_{ans}(h|u) + (1 - \lambda) * p_{ans}(u|\theta h)^k \quad (12)$$

## 4 Experiments

We crawled 40,000 resolved questions from Yahoo! Answers service in English for our experiments, and these questions from five categories respectively such as *Martial arts*, *Cycling*, *Health*, *Pets*, and *Software*, and there are 8,000 questions among each category. For each category, ten colleague students volunteer to be our assessors and each assessor also has interesting and basic knowledge at one or more categories. In general, we let assessors choose questions from each category randomly and label the question as ‘hard’ if they think that the question is hard to answer, otherwise, they label the question as ‘easy’ if they think the question is easy to answer. In this step, although people consider the question as easy or hard is subjective, we do our best to label it and omit the ambiguous questions, and Table 1 is our answer set for each category.

**Table 1.** The answer set for each categories

	Martial arts	Cycling	Health	Pets	Software
# Hard Q	93	75	70	70	70
# Easy Q	93	75	70	70	70

### 4.1 Baseline

We utilized five baseline methods in order to demonstrate the effectiveness of our approach:

1. Eigenrumour [4] algorithm: the linked based algorithm for ranking blogs, and the relationship between users and blogs in Eigenrumour is similar to the relationship between users and questions in our approach.
2. Base probability: the score that only do a half of KGB-DR, i.e., the iteration number is 1 in KGB-DR
3. Base Hits: there are two steps in this approach. First, calculate the auth score and hub score of users by [6]. Second, compute the score of the question by the sum of the hub score of asker and auth score of answerers.
4. #Words: the number of words in the question included answers
5. #Answerers: the number of answerers in the question

### 4.2 Evaluation Metrics

In order to evaluate the performance of our system, we use the four evaluation metrics such as the precision, the recall, the F-measure. In general, we set a variable “N” as a threshold, i.e., if the rank of the question is more than N, the question is regarded as a hard question, and while the question is regarded as an easy question instead. Furthermore, as same as the prior work [1], we utilize the above evaluation metrics for “hard question” and “easy question” separately, and both are measured when the experiment threshold N is set to maximize the F-measure. We will use these evaluating strategies to show how the performance of our methodology.

### 4.3 Comparison to Baseline

The final tuning of parameters for our two models is reported in Table 2 and the comparison of methods for each category is represented later.

**Table 2.** The parameters for the two models

	i	$\alpha$	$\beta$	$\lambda$	t
LKG-DRank	15	0.3	0.7	0.001	15%
GKG-DRank	10	0.4	0.6	0.4	15%

#### 4.3.1 F-measure of Easy Question and Hard Question

The performance of the methods for all test data sets with the F-measure of hard question is summarized in Table 3. The results presented in the table show that our approach with the LKG-DRank exhibits the best performance among all the other methods in detecting the hard question. The performance of the method we presented with the GKG-DRank is similar to the Base probability and the two methods show the second to the best performance. The #Answerers showed the worst performance and it shows that the more the answerers in the question, the easier the question becomes. Although EigenRumor also utilize the relationship between users and questions, the method shows poor performance due to the phenomenon of the experts can answer easy questions.

**Table 3.** The F-measure of hard question for each method and each category

	Martial arts	Cycling	Health	Pets	Software
LKG-DRank	<b>0.77</b>	<b>0.71</b>	<b>0.79</b>	<b>0.81</b>	0.50
GKG-DRank	0.67	0.70	0.66	0.68	0.55
Base probability	0.68	0.70	0.64	0.68	<b>0.59</b>
Base Hits	0.66	0.64	0.60	0.69	0.53
EigenRumor	0.50	0.54	0.61	0.64	0.41
#Words	0.67	0.60	0.55	0.53	0.38
#Answerers	0.48	0.41	0.50	0.46	0.31

Table 4 represents the performance of the methods for all test data sets with the F-measure of hard question. Compare with Table 3, the performance with all methods are increasing, but our approach with LKG-DRank also exhibits the best performance for all categories. It is reasonable that the part of the expert computed in KGS is higher than the part of the non-expert among all categories. In Table 4, the method that the performance is most increasing compared to Table 3 is #Answerers. It represents that the easy question is identified easier than the hard question via #Answerers.

**Table 4.** The F-measure of easy question for each method and each category

	Martial arts	Cycling	Health	Pets	Software
LKG-DRank	<b>0.78</b>	<b>0.73</b>	<b>0.81</b>	<b>0.83</b>	<b>0.69</b>
GKG-DRank	0.73	<b>0.73</b>	0.75	0.76	<b>0.69</b>
Base probability	0.75	<b>0.73</b>	0.75	0.76	<b>0.69</b>
Base Hits	0.70	0.72	0.71	0.71	0.68
EigenRumor	0.67	0.67	0.67	0.69	0.64
#Words	0.72	0.66	0.73	0.66	0.63
#Answerers	0.68	0.59	0.61	0.62	0.59

#### 4.3.2 The Examples of the Outputs Compared with the Basic Approaches

In order to compare the effectiveness of our approach, we give the particular query for searching questions in the particular category with cosine similarity. For example, we give the query “karate” in the category Martial arts and there are 358 questions in the output. For the purpose of more showing the effectiveness of our approach, we list the other questions and rank the questions with different methods and the outputs is shown in Table 5. The numbers in the brackets in Table 5 represent the corresponding methods. For example, the number 2,768 in No1 question represents there are 2,768 words in the question including asking and answering and the number 161 represents the order of the question is 161 of 358 by ranking with #words. The top three questions in Table 5

are detected as hard questions by our approach and the order of the rank is 36, 39, and 56 respectively. The score of the top three questions computed by our approach is higher and we can tell the three questions as hard by the question title, however, the order of the same questions is last by the other two methods due to the less words and number of answers in the questions. On the contrary, the last two questions in Table 5 are detected as easy question by our approach and it is easy to tell by the title of the questions. But the two easy questions attract some non-experts to answer it and it bring about the high order in the methods with number of words and answers. In addition, the question of No2 and the question of No4 are similar in the particular viewpoint, however; the different type of asking may make question be easy or hard. The question of No4 is the popular question of the amateur, but the question of No2 repeats asking the same thing by the different viewpoints and it would make this question hard. The two examples can prove the two similar questions may not be same degree of difficulty.

**Table 5.** The example of questions with query “karate” by our approach (358 total questions)

No.	Question title	Rank with our method	Rank with #words	#Answers	Degree of difficulty
1	would shotokan karate help for a mma career?	36	161(2768)	5	Hard
2	what age would you consider being to old to start karate?	39	255(1393)	8	Hard
3	Any karatekas who train with partial or minimal meniscus?	56	162(2757)	2	Hard
4	Starting karate at 16?	317	45(6133)	22	Easy

## 5 Conclusion

In this paper we defined the problem of detecting the question is easy or hard in YA and addressed it as a probability model and then utilize the phenomenon called knowledge gap in CQA service to solve this task. The contributions of this paper include:

1. We observe the unreasonable relationship between users from knowledge gap diagram, i.e., the expert can also answers the easy question in YA and it is not a good situation for the expert-finding via link analysis.
2. We present the approach called KG-DRank algorithm combining the relationship between users and the architecture in YA or CQA service, and the experiments shows the performance of our approach with LKG-DRank is the best among all baseline.
3. We demonstrate that the phenomenon of knowledge gap in our experiments and the performance reflect to the strength of the knowledge gap in the particular category.



## References

1. Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding high-quality content in social media. In: Proceedings of the International Conference on Web Search and Web Data Mining, pp. 183–194. ACM, Palo Alto (2008)
2. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. In: Proceedings of the Seventh International Conference on World Wide Web 7, pp. 107–117. Elsevier Science Publishers B. V., Brisbane (1998)
3. Bian, J., Liu, Y., Agichtein, E., Zha, H.: Finding the right facts in the crowd: factoid question answering over social media. In: Proceeding of the 17th International Conference on World Wide Web, pp. 467–476. ACM, Beijing (2008)
4. Fujimura, K., Inoue, T., Sugisaki, M.: The EigenRumor Algorithm for Ranking Blogs. In: WWW 2005 Workshop on the Weblogging Ecosystem 2005 (2005)
5. Fang, H., Zhai, C.: Probabilistic models for expert finding. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 418–430. Springer, Heidelberg (2007)
6. Jurczyk, P., Agichtein, E.: Discovering authorities in question answer communities by using link analysis. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 919–922. ACM, Lisbon (2007)
7. Jurczyk, P., Agichtein, E.: Hits on question answer portals: exploration of link analysis for author ranking. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 845–846. ACM, Amsterdam (2007)
8. Jeon, J., Croft, W.B., Lee, J.H., Park, S.: A framework to predict the quality of answers with non-textual features. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 228–235. ACM, Seattle (2006)
9. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *J. ACM* 46, 604–632 (1999)
10. Liu, X., Croft, W.B., Koll, M.: Finding experts in community-based question-answering services. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 315–316. ACM, Bremen (2005)
11. McCallum, A., Corrada Emmanuel, A., Wang, X.: Topic and role discovery in social networks. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, pp. 786–791. Morgan Kaufmann Publishers Inc., Edinburgh (2005)
12. Suryanto, M.A., Lim, E.P., Sun, A., Chiang, R.H.L.: Quality-aware collaborative question answering: methods and evaluation. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, Spain, pp. 142–151. ACM, Barcelona (2009)
13. Su, Q., Pavlov, D., Chow, J.-H., Baker, W.C.: Internet-scale collection of human-reviewed data. In: Proceedings of the 16th International Conference on World Wide Web, pp. 231–240. ACM, Banff (2007)
14. Zhang, J., Ackerman, M.S., Adamic, L.: Expertise networks in online communities: structure and algorithms. In: Proceedings of the 16th International Conference on World Wide Web, pp. 221–230. ACM, Banff (2007)
15. Zhou, Y., Cong, G., Cui, B., Jensen, C.S., Yao, J.: Routing Questions to the Right Users in Online Communities. In: Proceedings of the 2009 IEEE International Conference on Data Engineering, pp. 700–711. IEEE Computer Society, Los Alamitos (2009)