Information Processing and Management 49 (2013) 312–329

Contents lists available at SciVerse ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Integrating expert profile, reputation and link analysis for expert finding in question-answering websites ☆

Duen-Ren Liu *, Yu-Hsuan Chen, Wei-Chen Kao, Hsiu-Wen Wang

Institute of Information Management, National Chiao Tung University, Hsinchu 300, Taiwan

ARTICLE INFO

Article history:
Received 29 November 2009
Received in revised form 1 June 2012
Accepted 6 July 2012
Available online 9 August 2012

Keywords:
Community
Expert finding
Question answering
Link analysis
User reputation
Yahoo! Answer Taiwan

ABSTRACT

Question answering websites are becoming an ever more popular knowledge sharing platform. On such websites, people may ask any type of question and then wait for someone else to answer the question. However, in this manner, askers may not obtain correct answers from appropriate experts. Recently, various approaches have been proposed to automatically find experts in question answering websites. In this paper, we propose a novel hybrid approach to effectively find experts for the category of the target question in question answering websites. Our approach considers user subject relevance, user reputation and authority of a category in finding experts. A user's subject relevance denotes the relevance of a user's domain knowledge to the target question. A user's reputation is derived from the user's historical question-answering records, while user authority is derived from link analysis. Moreover, our proposed approach has been extended to develop a question dependent approach that considers the relevance of historical questions to the target question in deriving user domain knowledge, reputation and authority. We used a dataset obtained from Yahoo! Answer Taiwan to evaluate our approach. Our experiment results show that our proposed methods outperform other conventional methods.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Question answering websites, also known as Q&A websites, are becoming a more and more popular knowledge sharing platform. The major reason for this is not only because people can post natural language questions, but also because they can share miscellaneous information or obtain the answers to their questions directly from the website. In addition, people sometimes just need opinions so that they will be more likely to seek for help through question-answering websites. The Yahoo! Answer Taiwan website (http://tw.knowledge.yahoo.com/), also named Yahoo! Knowledge plus, is a community-driven knowledge website; each user can share experience and exchange knowledge by asking and answering questions. On the website, users can browse the questions that other users have asked, search for answers to particular questions, or post questions and then wait for answers. Every solved question has a "best answer". To choose a best answer, the asker can either select an answer as the best answer or set the question–answer pair to a voted stage. The answer that receives the highest vote is chosen as the best answer. Moreover, users can give any solved question an evaluation (positive, neutral, or negative) regarding whether the question–answer pair is useful or not.

Unfortunately, such question answering mechanisms may run into some issues. When the quantity of questions waiting to be solved grows quickly, some questions may be skipped by users who can answer, and askers will waste a lot of time to

☆ Extended version of "Expert Finding in Question-Answering Websites: A Novel Hybrid Approach". W.-C. Kao, D.-R. Liu, and H.-W. Wang. In *Proceedings of ACM Symposium on Applied Computing*, *SAC 2010*. Sierre, Switzerland.

* Corresponding author.
E-mail addresses: dliu@mail.nctu.edu.tw, dliu@iim.nctu.edu.tw (D.-R. Liu).

0306-4573/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ipm.2012.07.002

obtain answers. Even worse, askers may not obtain correct answers from appropriate experts. Consequently, knowledge sharing through question answering websites is interfered with such issues. Therefore, it is essential to automatically find appropriate experts for target questions, so as to shorten the waiting time, increase the quality of answers, and thus enhance the effectiveness of knowledge sharing.

Recently, various approaches have been proposed to automatically find experts in question answering websites. The link analysis approaches, including HITS (Kleinberg, 1999) and PageRank (Page, Brin, Motwani, & Winograd., 1998), have been adopted to find experts. Jurczyk and Agichtein (2007) adopt the HITS algorithm for author ranking. They represent the relationship of asker and answerer as a link network and calculate each user's hub and authority value, and then rank users according to their authority values. Zhang, Ackerman, and Adamic (2007) propose a PageRank-like algorithm called "ExpertiseRank" to rank experts in an expertise network considering how many people are involved and who has helped whom. Moreover, the user (expert) profiling approach (Liu, Croft, & Koll, 2005; Zhang, Ackerman, Adamic, & Nam, 2007), which built expert profiles from the contents of expert's questions and answers, is adopted to find experts without considering the reputations of experts and their authority values derived from link analysis.

In this paper, we propose novel methods to find appropriate experts to answer a given target question. A hybrid approach is proposed to effectively find experts for the category of the target question in question-answering websites. Different from the conventional approaches that only consider user profile or user authority, our approach considers user subject relevance, user reputation and authority of a category in finding experts. A user's subject relevance denotes the relevance of a user's domain knowledge to the target question. A user's domain knowledge for a specific category is represented as a user knowledge profile derived from the content and quality measures (e.g. voting/evaluation factor) of the user's historical question–answers in that category. A user's reputation in a category is derived from the user's historical question-answering records based on the ratio of the user's answers being adopted as best answers in that category, while user authority in a category is derived by applying link analysis to the category-based asker-answerer network.

In Yahoo! Answer, the range of a category domain is not small enough, so the domain experts may not be appropriate to answer the target question. We further extended our proposed category-based approach to develop a question dependent approach that considered the relevance of past questions to the target question in deriving user domain knowledge, reputation and authority. Our approach can enhance the quality of recommending experts through matching the content by relevance and taking the user reputation and user authority into account. The experiment result shows that our approach is better than other approaches that only use link analysis or user profile. Moreover, the question-dependent approach leads to a better result of finding experts for a target question.

The remainder of this paper is organized as follows. Section 2 presents the related literatures. Our proposed approach for expert finding is illustrated in Section 3. In Section 4, experiment evaluations are conducted to compare our approach with other methods. Conclusions and future research directions are finally presented in Section 5.

2. Related work

The widely used method to find the experts in community-driven question answering websites is link analysis or social network analysis. This method comprises of building a user social network first, and then using some kind of propagation algorithm to calculate each user's authority. Jurczyk and Agichtein (2007) build a link network based on the relation of asker and answerer between users in question–answer portals such as Yahoo! Answers. There exists an edge from user i to user j if user j has answered a question posted by user i. The link network is a multigraph in which multiple edges may exist between the two users if user j has answered several questions posted by user i. The authors do not consider the votes or best answers received by the answerer, and thus each edge is assigned the same weight. To discover the authorities in a particular category, the questions and their answers of that category are used to build the graph, and then the HITS algorithm is used to compute each user's authority value. Zhang, Ackerman, and Adamic (2007) address the issue of expert finding in an online help seeking community – the Java forums. An expertise (post-reply) network is constructed by viewing each user as a node and creating a directed edge from the user making the post to those users who replied to it. A PageRank-like algorithm called ExpertiseRank is adopted for expertise ranking. The edge of the network can be weighted according to the number of times one replies another. In addition, social network analysis has also been applied to derive users' reputations in a user-interactive question answering system – *CuiteAid* (Chen, Zeng, & Liu, 2006).

Besides the social network or link-based analysis, there are some other ways to find experts. Although link analysis methods have been adopted to find top-K users in a ranked list based on their expertise scores or authority values on subjects of interest, it is difficult to set appropriate values for K. Bouguessa, Dumoulin, and Wang (2008) address such issue and propose a probabilistic approach for automatic identification of authoritative actors in QA forums such as Yahoo! Answers. The proposed approach discriminate authoritative and non-authoritative users based on a mixture of gamma distributions. Zhang, Ackerman, and Adamic (2007) and Zhang, Ackerman, Adamic, and Nam (2007) address the issue of expert-finding in an online help seeking community – the Java forums. The vector space model in information retrieval (Manning, Raghavan, & Schütze, 2008) is used to represent the question and user profiles as term vectors. The proposed expert-finding method not only compares the similarity of questions and user profiles, but also considers the differences of expertise level, posting time of query, and the number of replies (status) to questions. Liu et al. (2005) adopt the language models in information retrieval (Manning et al., 2008) to build expert-profile models from the content of

question–answer pairs, and then ranks experts according to how likely the target question (query) could be generated from each of the expert-profile models.

The above papers mainly address expert finding in an open-domain community-based QA service such as Yahoo! Answers or Java forums. A lot of work has been done in the area of finding experts for a given topic from a domain-specific document collection or within an organization. Citation-based document retrieval systems have been enhanced for finding research domain expertise through context-based cluster analysis and ontology learning (Tho, Hui, & Fong, 2007). Moreover, the expert finding task has been included in the Enterprise track of the Text REtrieval Conference (TREC) (Voorhees, 1999). Demartini, Gaugaz, and Nejdl (2009) propose a general model for finding entities (experts) by extending the classical Vector Space Model to represent each entity as a weighted profile, i.e., a linear combination of document vector. The weight between an entity and a document indicates how much the entity is relevant to the subject addressed by a document. The well known cosine similarity measure can then be used to rank candidate experts based on their similarity measures to the query topic. Macdonald and Ounis (2008) propose a voting model for the application of expert search in the TREC collection of documents. The documents associated to a candidate expert are viewed as votes for this candidate's expertise. The authors also adopt a field-based weighting model, which assigns different weights to three fields – the body, the title, and the anchor text of incoming hyperlinks of document, to improve the ranking of candidates. The voting model has also been used to integrate additional evidence by identifying candidate homepages and clustering candidate profiles for expert search (Macdonald, Hannah, & Ounis, 2008).

Moreover, several methods for expert search in the TREC collection of documents have been proposed based on the language-modeling techniques to estimate the probability of the query topic being generated by the candidate expert. There are two approaches to expert finding (Balog, Azzopardi, & de Rijke, 2009; Fang & Zhai, 2007; Serdyukov, Rode, & Hiemstra, 2008). The profile-based methods build a term-based representation for each candidate, and rank the candidate experts based on the relevance scores of their profiles for a given topic. The document-based methods first find the supporting documents relevant to the topic and then rank the candidates according to the sum of relevance probabilities of the supporting documents. Balog and de Rijke (2007) propose profiling methods with filtering to determine a candidate's expert profile, which is expressed as a vector of knowledge areas and associated competency of the candidate. Two profiling methods are proposed, one is based on the language modeling techniques and the other is based on the TF-IDF weighting approach. Moreover, Balog et al. (2009) propose a general probabilistic framework for expert finding in the context of enterprise search systems within an intranet environment. The proposed framework is based on the language modeling techniques and considers various expertise search strategies. Serdyukov and Hiemstra (2008) propose a mixture of personal language models for expert finding. Serdyukov et al. (2008) model candidate experts (persons), web documents and their relations as expertise graphs, and propose expert finding methods based on multi-step relevance propagation in topic-specific expertise graphs.

Finding high quality answers in question answering websites is another popular issue. Basic studies use statistics to find the content factors that may influence the selection of best answers (Kim, Oh, & Oh, 2007). More thorough studies have considered non-textual and textual features to identify high quality answers (Bian, Liu, Agichtein, & Zha, 2008; Blooma, Chua, & Goh, 2008; Jeon, Croft, Lee, & Park, 2006). Suryanto, Lim, Sun, and Chiang (2009) propose a quality-aware framework that considers both answer relevance and answer quality derived from answer features and answerers' expertise.

3. Proposed mechanism for expert finding

3.1. Framework for expert finding

Our proposed methods combine user knowledge profiles, user reputations and link analysis to find experts for a given category or target question. Fig. 1 shows our proposed framework for expert finding. The historical question–answer data of users is collected and analyzed using information retrieval techniques to build up user knowledge profiles, which represent the knowledge subjects of users. We gather the content of question–answer pairs and information related to question–answer pairs from the community question answering website, including posting time, category, and so on. We use the vector space model (Manning et al., 2008) to process the content of question–answer pairs, which will be converted to term vectors using the TF-IDF approach (Salton & Buckley, 1988) at data preprocessing stage. In differentiating our approach from the conventional methods, we consider the category of question–answer pair and assign different priorities (weights) to terms according to the place (question title, question description or answer) of the term in deriving the term vectors. User knowledge profiles are derived from users' related question–answer pairs, which are expressed in term vectors. Moreover, we not only analyze the content of historical question–answers but also their relevance to the target question, their quality measures (derived from the voting/evaluation factor) and time factor, as weight to build up user knowledge profiles. Conventional approaches derive user profiles without considering the quality measures and the relevance of past questions to the target question.

We use a user's knowledge profile to derive a user's knowledge score by comparing the similarity of the profile and the target question. In addition, a user's reputation score is derived according to users' historical question-answering records, including the number of answers and best answers given by users. The link analysis approaches, including HITS and PageRank, are used to find the authority scores of users based on the link network created from the links that connect users who give answers to users who ask questions. In the expert recommendation phase, a hybrid method that combines a user's

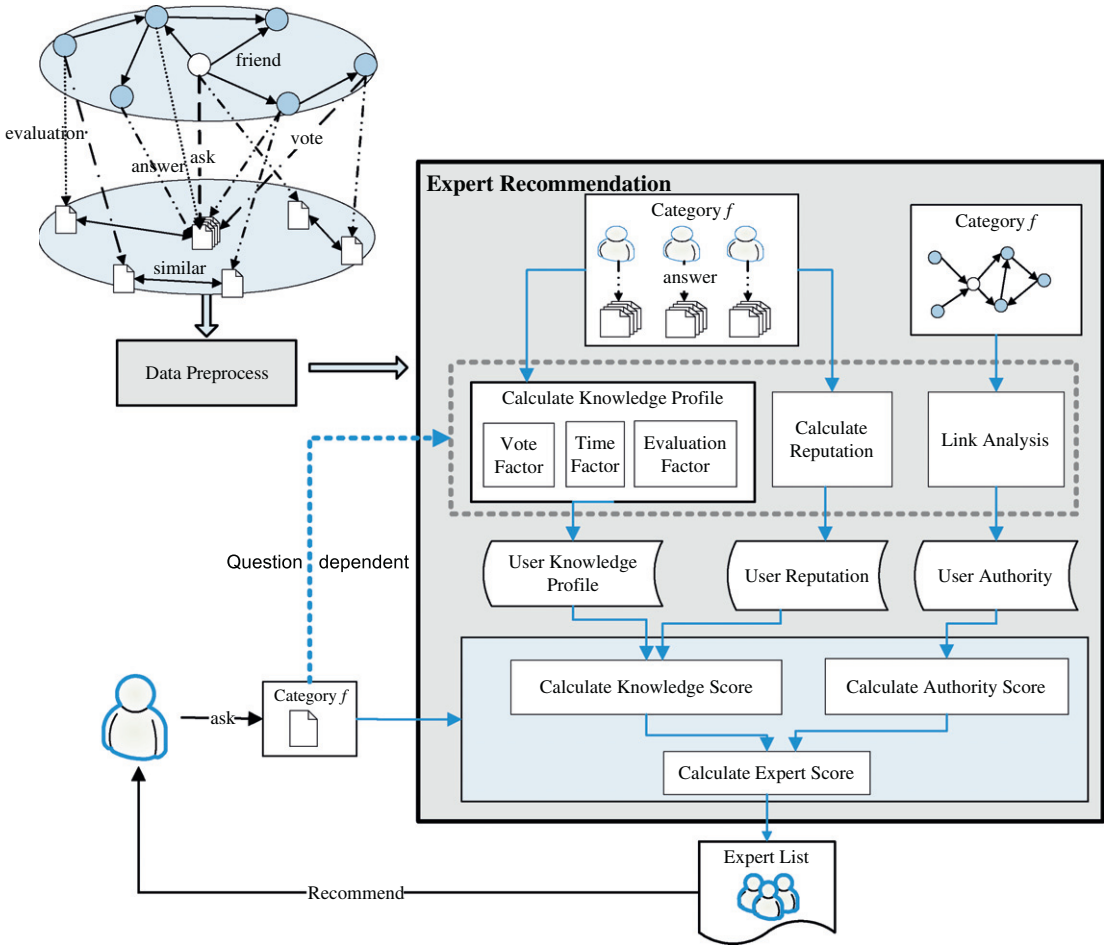

Fig. 1. Proposed framework for expert finding.

knowledge score, reputation score and authority score is used to derive a user's expert score and produce a recommended expert list.

Two approaches, a category-based approach and a question-dependent approach, are proposed to find experts for a given target question of a category. The category-based approach derives the category-based user knowledge profiles, reputation scores and authority scores from historical question-answering records of a category without considering the relevance of past questions to the target question. We further extend the category-based approach to develop a question dependent approach (the dotted line in Fig. 1) that considers the relevance of past questions to the target question in deriving user knowledge profiles, reputation scores and authority scores. The details are presented in the following subsections.

3.2. Data preprocessing

The data was collected from the famous question-answering website – Yahoo! Answers – in Taiwan. The relation of question and answer in Yahoo! Answers is one to multiple. In other words, one question may have more than one answer, where we define that as a "question-all-answers pair" (QAA pair). The goal is to build up each user's knowledge profile by their own answering records in the past, so we separate all answers of a question by users as a user question–answer pair (UQA pair).

The content of question title, question description and answer in each UQA pair is analyzed using the TF-IDF approach to extract the important terms that can represent the knowledge subjects of the UQA pair. The data pre-process includes CKIP (Chinese Knowledge and Information Processing; http://ckip.iis.sinica.edu.tw/CKIP/engversion/index.htm), the removal of stop words, and the extraction of TF-IDF for each term. CKIP provides a Chinese parser functionality to facilitate word segmentation and derive the morphological information of each word. TF-IDF (Salton & Buckley, 1988) is used to calculate the weight of term i in a UQA pair j. It can be calculated for a given category or a whole data set, as defined in the following equation:

$$w_{i,j} = f_{i,j} \times \log\frac{N}{n_i}; \quad f_{i,j} = \frac{freq_{i,j}}{\max_l(f_{l,j})}, \tag{1}$$

where N is the number of user question–answer (UQA) pairs in a given category or whole data set; n_i is the number of UQA pairs that contain term i, in a given category or whole data set; $f_{i,j}$ is the normalized frequency of term i in UQA pair j; $freq_{i,j}$ is the frequency of term i in UQA pair j; $\max_l(f_{l,j})$ is the frequency of term l that has the maximum frequency in UQA pair j.

In this step, we use a term vector to represent the knowledge subjects of a UQA pair that consists of the question title, question description, and answer fields. There are two approaches to derive the term vector. One is deriving term weights according to the term frequency occurring in the UQA pair without considering the field weight. Alternatively, we adopt a field-based weighting approach, which is similar to the idea presented in (Macdonald & Ounis, 2008), to give a term different importance (weights) according to the field (e.g. question title, question description or answer) where the term occurred.

We derive the weights of question title, question description, and answer via a weighting scheme that is defined based on the evaluation metric, MRR and precision. The approach is described as follows. The term vector of each field i.e. question title, question description, and answer of a UQA pair is used to retrieve similar UQA pairs, respectively. Our data set contains training, analytical and testing datasets. The analytical dataset consisted of 100 questions is used to determine various factors and the values of parameters in our proposed methods. We randomly chose 20 questions from the 100 analytical questions to search relevant QAs from our training data set. The correctness of the top ten retrieval results, which is judged by human experts, is used to calculate the weight score of each field respectively, as in the following equation:

$$FW = \frac{1}{r_i} \times \frac{n}{m}, \tag{2}$$

FW is the weight score of a field, i.e. question title, question description or answer; n is the number of correct UQA pairs; m is the number of UQA pairs retrieved; r_i is the rank of the first correct UQA pair occurred. We average the weight scores of a field for a given set of UQA pairs to derive $FW(T)/FW(D)/FW(A)$, the average weight score of the question title (T)/description (D)/answer field (A). The average weight score $FW(T)$ is divided by the summation of $FW(T)$, $FW(D)$ and $FW(A)$ to derive the normalized weight scores of the title field ω_{title}. The normalized weight scores of the question description $\omega_{question}$ and the answer field ω_{answer} are derived similarly. Consequently, the weighted term weights can be derived according to Eq. (3), where $w^T_{i,j}/w^D_{i,j}/w^A_{i,j}$ is the weight of term i occurred in the title/question description/answer field of the UQA pair j (Eq. (1)). We can get four kinds of term vectors for a UQA pair: derived from a given category/whole data set and with/without considering field weight.

$$w_{i,j} = \omega_{title} \times w^T_{i,j} + \omega_{question} \times w^D_{i,j} + \omega_{answer} \times w^A_{i,j}. \tag{3}$$

3.3. Build up of user knowledge profiles by category

In order to identify those experts who are appropriate to answer the target question, the knowledge subjects of users, which are derived from users' historical question-answering records, will be an important factor. We build up the users' knowledge profiles expressed as term vectors to represent users' knowledge subjects.

Let $UQA^{q_r}_{u_a}$ denote a user-question–answer pair that contains question q_r and its answer given by user u_a. QAA^{q_r} denotes a question q_r and all its answers. The knowledge profile of user u_a can be derived from all the UQA pairs of user u_a. Each UQA pair may have different importance in deriving the knowledge profile of a user. We weight each UQA pair by three factors: the vote factor V_{u_a,q_r} derived from the votes given to user u_a's answer for question q_r, the evaluation factor E_{q_r} of question q_r, and time factor T_{q_r} of question q_r. Each user may have more than one subject domain; thus, we derive the knowledge profiles by subject category. The knowledge profile of user u_a in category f, defined as $KP^f_{u_a}$, is derived according to the following equation:

$$KP^f_{u_a} = \frac{\sum_{q_r \in C^f_{u_a}} AP_{u_a,q_r} \times V_{u_a,q_r} \times E_{q_r} \times T_{q_r}}{|C^f_{u_a}|}, \tag{4}$$

where $C^f_{u_a}$ is the set of questions belonging to category f and answered by user u_a; q_r is the question that user u_a answered; AP_{u_a,q_r} is the term vector of $UQA^{q_r}_{u_a}$, i.e. the question q_r and its answer given by user u_a; $|C^f_{u_a}|$ is the number of questions answered by user u_a in category f. The three factors will be explained in the following.

In the Yahoo! Answers website, an asker can choose one answer as the best answer or put the answers into a voting stage. The answer getting the most votes will be the best answer. We consider that the more votes the answer gets, the more important it is. The vote factor V_{u_a,q_r} of a UQA pair for question q_r and its answer given by user u_a is derived as follows:

$$V_{u_a,q_r} = \frac{\text{vote to } UQA^{q_r}_{u_a} + \mu}{\#vote\ to\ QAA^{q_r} + |QAA^{q_r}| \times \mu}, \tag{5}$$

where $|QAA^{q_r}|$ is the number of answers for question q_r. The parameter μ is an adjusting value to avoid zero values for answers that do not get any votes. In our experiment, we set $\mu = 0.1$ to avoid ignoring the UQA pair when it did not receive any

votes. If the answers do not put into the voting stage, i.e., the best answer is selected by the asker, we use Eq. (6) to derive the value of V_{u_a,q_r}. The best answer will get θ, which is set as 0.6 to ensure that the best answer gets the maximum value among the answers of the same question.

$$V_{u_a,q_r} = \begin{cases} \theta & \text{If best answer} \\ \frac{1-\theta}{|QAA^{q_r}|-1} & \text{Otherwise} \end{cases}. \tag{6}$$

Furthermore, the evaluation made by users is another approximate basis to discriminate the importance of UQA pairs. If someone thinks the question (and answer) is useful/not useful, he can give it a positive/negative evaluation. Note that a user evaluates the whole QAA, i.e. a question and all its answers, rather than a UQA pair. This evaluation may be positive, neutral or negative. We consider that the positive evaluation is plus, the neutral is zero, and the negative one is minus, as defined in Eq. (7). To prevent the value of evaluation becoming zero, we add one to the result.

$$E_{q_r} = 1 + \frac{\#\text{ positive evaluation} - \#\text{ negative evaluation}}{\#\text{ evaluation}} \tag{7}$$

The last factor we take into account is the time factor. It is reasonable to assume that the knowledge in a UQA pair posted in the recent past is more important and up-to-date; that is to say, the older user-question–answer pairs should be given lower importance. We consider the time decay effect of the UQA pairs. Each UQA pair is assigned a time weight according to the time it was posted. Thus, higher time weights are given to UQA pairs posted in the recent past. The time weight of each UQA pair is defined as Eq. (8). We adopt the formula in (Zhang, Ackerman, Adamic, & Nam, 2007) to compute our time factor.

$$T_{q_r} = e^{-\tau(t_{now}-t_{q_r})}, \tag{8}$$

where t_{now} is the present date time; t_{q_r} is the date time of the question q_r being posted; τ is the tunable parameter, which we set at 1/365 to avoid dropping too fast.

3.4. Expert recommendation

In the expert recommendation phase, a hybrid method that combines a user's knowledge score and authority score is used to derive a user's expert score and produce a recommended expert list. A user's knowledge score represents a user's knowhow and reputation on the subjects related to the target question. We use a user's knowledge profile to derive a user's knowledge score by comparing the similarity of the profile and the target question. In addition, a user's reputation is derived according to users' historical question-answering records, including the number of answers and best answers given by users. The authority score represents a user's authority on a social network created from the question–answer relationships between users. The link analysis approaches, including HITS and Page Rank, are used to find the authority scores of users based on the link network created from the links connecting users who give answers to questioners.

A user u_a's expert score $ExpertScore_{u_a,q}$ for a given target question q is derived as Eq. (9). The equation is composed of two parts. One is the knowledge score defined as $K\_score_{u_a,q}$, which is calculated from user knowledge profile and user reputation. The other one is the authority score defined as $A\_score_{u_a,q}$, which is calculated from the link analysis.

$$ExpertScore_{u_a,q} = \beta \times K\_score_{u_a,q} + (1-\beta) \times A\_score_{u_a,q}, \tag{9}$$

where β is the parameter used to adjust the relative importance of knowledge score and authority score. The derivations of a user's knowledge score and authority score are illustrated in Section 3.4.1 and 3.4.2, respectively.

After calculating all users' expert scores on the target question, we sort the expert score with a descending order, and the top K users will be our recommended experts.

3.4.1. The knowledge score

Let f be the category of the target question q. Let $KP^f_{u_a}$ be the user u_a's knowledge profiles in category f and QP_q be the target question q's profile, which is a term vector extracted from the title and description of question q. The knowledge score $K\_score_{u_a,q}$ is calculated by considering the cosine similarity of $KP^f_{u_a}$ and QP_q, i.e., $sim(KP^f_{u_a}, QP_q)$, and the user u_a's reputation score in category f, $Reputation^f_{u_a}$, as shown in Eq. (10).

$$K\_score_{u_a,q} = \alpha \times sim(KP^f_{u_a}, QP_q) + (1-\alpha) \times Reputation^f_{u_a}, \tag{10}$$

where α is a parameter used to adjust the relative importance of u_a's knowledge profile and reputation score. $sim(KP^f_{u_a}, QP_q)$ denotes the relevance of user u_a's subject knowledge to the target question. A user's knowledge profile, $KP^f_{u_a}$, is derived according to Eq. (4). The calculation of user's reputation score is illustrated in the following.

3.4.1.1. Reputation sore. We compute a user u_a's reputation, $Reputation^f_{u_a}$, based on category reputation, $NCR^f_{u_a}$. The basic concept of category reputation is to calculate the ratio of best answers to answers given by user u_a, i.e. the adoption ratio of best answers in a given category. However, one may have high adoption ratio of best answers, but have answered very few questions. Thus, the number of best answers given by a user is also an important factor to derive a user's reputation. If a user has a higher adoption ratio of best answers and has a greater number of best answers, he/she has a higher reputation, as shown in

Eq. (11). The first part of the equation calculates the normalized adoption ratio of best answers, while the second part derives the normalized number of best answers of user u_a. We use a parameter λ $(0 \leqslant \lambda \leqslant 1)$ to adjust the relative importance of adoption ratio and the number of best answers. The rationale was to make sure that the reputation would not drop too fast when the adoption ratio of best answers or the number of best answers drop.

$$NCR^f_{u_a} = \frac{CategoryAdoptedRatio^f_{u_a}}{\max_{u_x} CategoryAdoptedRatio^f_{u_x}} \times \left[\lambda + (1-\lambda) \times \frac{BestAnswer^f_{u_a}}{\max_{u_x} BestAnswer^f_{u_x}}\right], \tag{11}$$

where $BestAnswer^f_{u_a}$ is the number of best answers in category f given by user u_a; $CategoryAdoptedRatio^f_{u_a}$ is the user u_a's adoption ratio of best answers in category f, which is derived by Eq. (12).

$$CategoryAdoptedRatio^f_{u_a} = \frac{\#\text{ best answer in category } f \text{ given by user } u_a}{\#\text{ answer in category } f \text{ given by user } u_a}. \tag{12}$$

3.4.2. The authority score

The authority score represents a user's authority in a link network created from the question-answering relationships between users. We use the links connecting users who give answers to questioners in a specific category to construct the graph. The graph is simplified into a multi-edges graph, where each node represents a user, and every edge represents the asking and answering relationship. The start node of an arrow is an asker who asks a question, and the end node of an arrow is an answerer who gives an answer to the question. A user may ask questions and answer questions posted by other users. Thus, a link network (asking-answering graph) is formed based on the question-answering relationships between users.

After constructing the asking-answering graph for category f, which the target question q belongs to, we need to calculate the *authority score*. There are two approaches to calculate the $A\_score_{u_a,q}$. One is the HITS algorithm (Kleinberg, 1999), which is shown in Eq. (13). $H(u_b)$ denotes user u_b's hub value, and $A(u_a)$ denotes user u_a's authority value.

$$H(u_b) = \sum_{u_a:u_b \to u_a} A(u_a);\ A(u_a) = \sum_{u_b:u_b \to u_a} H(u_b). \tag{13}$$

Every node has a hub value and an authority value. If one has a high hub value it means that he is a good asker, and if one has a high authority value it means that he is a good answerer. For the HITS approach, we use the authority value as our *authority score*.

The other approach is the PageRank algorithm (Page et al., 1998), which is shown in Eq. (14). We use the PageRank score as our *authority score*.

$$PR(u_a) = c \sum_{u_b:u_b \to u_a} \frac{PR(u_b)}{O(u_b)} + (1-c)\frac{1}{N}, \tag{14}$$

where $PR(u_a)$ denotes user u_a's *PageRank* score; $O(u_b)$ denotes user u_b's outdegree; c is the damping factor, in which we set at 0.85; and N is the total number of users.

3.5. Question-dependent expert finding

In previous subsections, user knowledge profiles and user reputations are derived for a given category. The category-based knowledge profiles and reputations can be derived in advance, and then be used to calculate expert scores when a query, i.e. target question, is submitted. The category-based approach can be extended to generate question-dependent knowledge profiles, reputations and link networks, which may lead to a better result when it comes to finding question-related experts. In this section, we present the formulas to generate user knowledge profiles, user reputations and link networks for target questions.

For building user knowledge profiles, we add the cosine similarity of the target question q and the question q_r in the Eq. (4), as listed in the following equation:

$$KP^f_{u_a}(q) = \frac{\sum_{q_r \in C^f_{u_a}} AP_{u_a,q_r} \times V_{u_a,q_r} \times E_{q_r} \times T_{q_r} \times sim(QP_q, QP_{q_r})}{|C^f_{u_a}|}, \tag{15}$$

where $sim(QP_q, QP_{q_r})$ is the cosine similarity between the target question q's profile and question q_r's profile. In calculating user reputations, we also add the cosine similarity of the target question q and the question q_r in the original Eqs. (11) and (12), as listed in Eqs. (16)–(18).

$$NCR^f_{u_a}(q) = \frac{CategoryAdoptedRatio^f_{u_a}(q)}{\max_{u_x} CategoryAdoptedRatio^f_{u_x}(q)} \times \left[\lambda + (1-\lambda) \times \frac{BestAnswer^f_{u_a}(q)}{\max_{u_x} BestAnswer^f_{u_x}(q)}\right]. \tag{16}$$

$CategoryAdoptedRatio^{f}_{u_a}(q)$ is the user u_a's adoption ratio of best answers for answering questions in category f, which is calculated by considering the questions' similarities to the target question q. $BestAnswer^{f}_{u_a}(q)$ is the number of best answers given by user u_a for answering questions in category f, which is also calculated by considering the questions' similarities to the target question q.

$$CategoryAdoptedRatio^{f}_{u_a}(q) = \frac{\sum_{q_r \in C^{f}_{u_a}} sim(QP_q, QP_{q_r}) IsBest(q_r, u_a)}{\sum_{q_r \in C^{f}_{u_a}} sim(QP_q, QP_{q_r})}, \tag{17}$$

$$BestAnswer^{f}_{u_a}(q) = \sum_{q_r \in C^{f}_{u_a}} sim(QP_q, QP_{q_r}) IsBest(q_r, u_a), \tag{18}$$

where $C^{f}_{u_a}$ is the set of questions belonging to category f and answered by user u_a; $IsBest(q_r, u_a)$ is a Boolean function; if user u_a's answer is the best answer for question q_r, the value will be 1; otherwise, the value will be 0.

Similarly, the link analysis can also be question-dependent. The links of the asking-answering graph can be weighted by the similarity of the target question q to the question q_r, as shown in Fig. 2.

We can then calculate the $A\_score_{a,q}$ by a weighted HITS with the Eq. (19).

$$\begin{aligned} H(u_b, q) &= \sum_{u_a : u_b \to u_a} sim(QP_q, QP_{q_r}) \times A(u_a, q) \\ A(u_a, q) &= \sum_{u_b : u_b \to u_a} sim(QP_q, QP_{q_r}) \times H(u_b, q) \end{aligned} \tag{19}$$

$sim(QP_q, QP_{q_r})$ is the cosine similarity between target question q's profile and question q_r's profile, where question q_r is asked by user u_b and answered by user u_a.

4. Experiment evaluations

Our dataset is collected from Yahoo! Answers in Taiwan. Two hundred questions and corresponding answers were randomly selected from four categories: "general disease", "medicinal usage", "traditional medicine", and "health information" during July 2007 to November 2008, where 100 questions and corresponding answers were used as the analytical data and the remaining 100 questions were used as the testing questions. There are 136 users who had answered the selected 200 questions. We used the 136 users as candidate experts, and collected their answering records including the questions they have answered, best answers, evaluations, their answers and corresponding votes. We note that the collected answering records might include questions they answered after the 200 analytical/testing questions. In addition, we collected the candidate experts' personal information, including knowledge file, knowledge grade, and knowledge circle such as "friend" and "fan". Table 1 lists the statistics of 136 users' answering records collected from January 2006 to November 2008. Expanded from the answering records of the 136 candidate experts, our dataset accumulated 52,899 questions and 215,504 corresponding answers. Our dataset excluding the analytical and testing data (the 200 questions and corresponding answers) is used as the training dataset to derive user knowledge profile, reputation score and authority score. The analytical dataset consisted of 100 questions is used to determine the factors in deriving knowledge profiles and the values of parameters – λ, α and β in calculating the reputation score (Eq. (11)), knowledge score (Eq. (10)) and expert score (Eq. (9)) respectively, as described in Sections 4.2-4.5. Once the factors and settings of parameters have been determined, we use the testing dataset to evaluate and compare our proposed approaches with conventional methods. The details are illustrated in Sections 4.6, 4.7 and 4.8.

Although the test data contains who of the 136 users provided the best answer to each question. There might be some users, who would have answered the question even better than the users that did answer the question. Since there is no standard result indicating who is appropriate to answer the target question, the evaluation is conducted by three human raters to justify manually the suitable experts for the target question. Every human rater judges the 136 candidate experts to determine the suitable experts for each of the 100 testing questions separately. For each of the three human raters, we

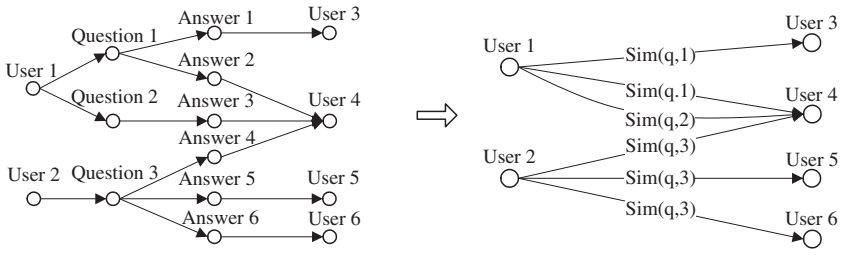

Fig. 2. Weighted asking-answering graph.

Table 1
The statistics of the users in the experiments.

Total num. of answer	Knowledge grade						
	Knowledge superior	Master	Expert	Postgraduate	Practicer	Novice	Sum
More than 1000	4	4	2	1	13	0	24
500–1000	0	5	4	6	6	1	22
100–500	0	0	15	16	20	18	69
Less than 100	0	0	0	0	0	21	21
Sum	4	9	21	23	39	40	136

calculate the number of testing questions for the rater to choose the user who provided the best answer to a testing question as a suitable expert in regard to the question, thereby yielding three outcomes. We then calculate the average of the three outcomes. On average, for 90% of the testing questions, the human raters choose the user who also provided the best answer to the question. We also calculate the average number of testing questions per human rater for choosing users who answered the questions but did not give the best answer. On average, for 77% of the testing questions, the human raters choose users who also answered the questions but did not give the best answer. On average, there are 8 experts per question chosen by the human raters. We analyze the agreement between the three human raters. For each testing question, we calculate the number of common experts chosen by all three human raters. We then calculate the average number of common experts over 100 testing questions. The average number of common experts chosen by all three human raters is 4.76 per testing question. The agreement between two human raters is also analyzed. The average number of common experts chosen by two human raters is 6.09 per testing question.

We use the Mean Reciprocal Rank (MRR), Precision at 5 (P@5) and Mean Average of Precision (MAP) as our evaluation metrics. MRR, Precision and MAP are widely used metrics to evaluate the effectiveness of retrievals to queries (Manning et al., 2008). The metrics have also been used in the expert search literature to evaluate the effectiveness of finding experts for questions/queries (Balog et al., 2009; Macdonald & Ounis, 2008). MRR calculates the reciprocal of the rank of the first suitable/relevant expert/item. Precision at K is the fraction of the top K experts/retrievals that are relevant to the question/query. MAP not only considers the percentage of relevant experts/items, but also takes the ranking positions of relevant experts/items into account.

The evaluation obtained by just using the best answerer of each question as the suitable expert of the question is limited, since there are some other users who are suitable experts to answer the question. The evaluation based on human raters also has a limitation: the justification of suitable experts for the target questions is subjective to each human rater's personal judgment. Different human raters may have different results. Accordingly, we calculate the MRR, P@5, and MAP of various methods by comparing their expert findings with three expert findings judged by human raters, thereby obtaining three outcomes. We use the average of the three outcomes as the final evaluation result. Taking the average of the three outcomes can reduce the subjective bias of human raters.

4.1. Experiment design

Our proposed hybrid methods combined a user's knowledge score and authority score to derive a user's expert score. Specifically, a user's knowledge score was derived by combining user reputation and the similarity between the user's knowledge profile and the target question's profile. A user's knowledge profile was derived by aggregating the term vectors of the user's past question–answer pairs. Different factors when extracting the term vectors and aggregating the term vectors may affect the quality of expert finding. While deriving the user's knowledge profile, the *tf-idf* approach was used to calculate the weight of terms and it could be calculated within a given category or a whole data set; the weight of field (question title/question description/answer) may also affect the result. In addition, the voting factor, time factor and evaluation factor were used as the weights of the term vectors to adjust the aggregations of the term vectors in deriving the knowledge profiles. We evaluated the effect of those factors on expert finding to determine the appropriate factors for deriving user knowledge profiles.

Despite considering the similarity between user knowledge profiles and the target question's profile, user reputation is also considered as another means of judging user expertise. Hence, we evaluated the effect of combining the two sources as the user's knowledge score for expert finding. Finally, the user's authority score calculated by link analysis was combined with the knowledge score to derive a hybrid effect of improving the quality of expert finding. We also conducted experiments to evaluate the hybrid effect.

Our hybrid approaches were category-based and question-dependent. The category-based approach derives user knowledge profiles, reputation and authority scores based on the past question-answering records of the given category to which the target question belongs. The question-dependent approach extends the category-based approach by further considering the relevance of past questions to the target question in deriving user knowledge profiles, reputation and authority scores. We, therefore, compared both category-based and question-dependent methods to verify the effect of considering the relevance of past questions to the target question on expert finding.

There were numerous combinations of factors, parameter settings and components of our proposed methods that would affect the quality of expert finding, and exploring all these combinations would have been difficult. Thus, we used a greedy strategy to derive our proposed methods by considering one more component at each step. We chose the factors and parameter settings which seemed best at the current step depending on the choices made thus far. That is, we made one greedy choice after another, without reconsidering previous choices. Based on this greedy strategy, we were able to conduct feasible experiments to choose and verify the factors and parameter settings of our proposed methods for expert finding.

The experiments formed the basis for a discussion of the issues listed below:

- What are the effects of different approaches on extracting the term vectors for deriving knowledge profiles? (Section 4.2)
- What are the effects of voting factor, time factor and evaluation factor on deriving knowledge profiles to improve the quality of expert finding? (Section 4.3)
- Does the method which considers user reputation for deriving user expertise (knowledge score) perform better than the one which does not consider user reputation? (Section 4.4)
- What is the effect of different link-based approaches on expert finding? What is the relative importance of user expertise (knowledge score) and the authority score of the hybrid approach for expert finding? (Section 4.5)
- What is the effect of different category-based methods on expert finding? Does the proposed category-based method perform better than other methods? (Sections 4.6 and 4.8)
- What is the effect of considering the relevance of past questions to the target question on expert finding? What is the effect of different question-dependent methods on expert finding? Does the proposed question-dependent method perform better than other methods? (Sections 4.7 and 4.8)

We compared various methods to evaluate the effectiveness of our proposed approaches. The baseline methods included, *KGrade*, *ExpertHITS*, *ExpertPRank, VSM*, and *CBDM*. *KGrade* ranks experts by their "knowledge grade"; and if they have the same knowledge grade, the method ranks them by the number of questions that they had answered. *ExpertHITS* and *ExpertPRank* are link-analysis methods for expert finding by using the HITS and PageRank algorithm, respectively. Zhang, Ackerman and Adamic (2007) adopt PageRank to measure a user's expertise in an expertise network. Their proposed approach, namely ExpertiseRank algorithm is similar to PageRank, and thus is also called a PageRank-like algorithm. In addition, they also adopt HITS to measure a user's expertise, where the authority value of HITS corresponds to the expertise rank of the user. Since the PageRank and HITS had been adopted to rank experts in an expertise network (Zhang, Ackerman and Adamic, 2007), we thus adopt *ExpertHITS* and *ExpertPRank* as the baseline methods for expert finding.

We use the training dataset to derive the expert scores of *ExpertHITS* and *ExpertPRank* methods. For each question in the training dataset, the asker and the corresponding answerers are the user nodes in the link network (asking-answering graph) created based on the question-answering relationships between users. We note that some users had both asked and answered questions in the training dataset. Accordingly, the PageRank scores of candidate experts depend on the PageRank scores of the users who asked the questions the candidate experts had answered.

VSM is a conventional vector space model based on the TF-IDF weighting approach. Liu et al. (2005) adopted a cluster-based document model (*CBDM*) (Liu and Croft, 2004) proposed for document retrieval to find experts in community-based question-answering services. A brief description of the *CBDM* method has been given, but readers may refer to references (Liu & Croft, 2004; Liu et al., 2005) for the full details. The expert profile, which consists of question–answer (QA) pairs derived from previous answered questions, is used to denote the expertise of a user. The expert profiles can then be ranked by using *CBDM* that considers the given target question as query and the expert profiles as documents. Let D denote an expert profile and *Coll* be the entire collection. Let *Cluster* denote the cluster of the expert profile derived from grouping expert profiles into clusters. Let w_i be the ith term in the target question/query. The question/query is treated as a sequence of independent terms. The query probability can be represented as a product of the individual term probabilities as defined in Eq. (20).

$$P(Q|D) = \prod_{i=1}^{m} P(w_i|D), \tag{20}$$

where $P(w_i|D)$ is defined in Eq. (21).

$$P(w|D) = \lambda P_{ML}(w|D) + (1-\lambda)P(w|Cluster) = \lambda P_{ML}(w|D) + (1-\lambda)[\beta P_{ML}(w|Cluster) + (1-\beta)P_{ML}(w|Coll)] \tag{21}$$

where $P_{ML}(w|D), P_{ML}(w|Cluster), P_{ML}(w|Coll)$ are the maximum likelihood estimates of word w in the document, cluster, and collection, respectively. Both $\lambda \in [0,1]$ and $\beta \in [0,1]$ are smoothing parameters, which can take the form as in Eq. (22) for Bayesian smoothing with the Dirichlet prior.

$$\lambda = \frac{\sum_{w' \in D} tf(w', D)}{\sum_{w' \in D} tf(w', D) + \mu}; \quad \beta \frac{\sum_{w' \in Cluster} tf(w', Cluster)}{\sum_{w' \in Cluster} tf(w', Cluster) + \mu}. \tag{22}$$

In Liu and Croft (2004), documents (expert profiles in this study) were clustered using a cosine similarity measure. Each document was smoothed with the cluster containing the document by interpolating the original maximum likelihood estimate,

$P_{ML}(w|D)$ with a cluster language model $P_{ML}(w|Cluster)$, which was further smoothed by interpolating itself with $P_{ML}(w|Coll)$.

The baseline methods compared in our experiments were as follows.

CBDM is the approach proposed by Liu et al. (2005).
VSM is a conventional vector space model based on the TF-IDF weighting approach. VSM builds up user profiles by aggregating UQA term vectors of questions answered by users in a given category. The user profile is derived according to the equation, as Eq. (4), without the V/E/T factors. VSM ranks experts according to the cosine similarity of user profiles and the target question.
ExpertHITS ranks experts based on their authority scores calculated by Eq. (13).
ExpertPRank ranks experts based on their PageRank scores calculated by Eq. (14).
KGrade ranks experts by their "knowledge grade"; and if they have the same knowledge grade, the method ranks them by the number of questions that they had answered.
QD-HITS ranks experts based on their authority scores calculated by Eq. (19), where the links of the asking-answering graph are weighted by the relevance of past questions to the target question.

Our proposed methods included several variations.

KProfile ranks experts based on the cosine similarities between users' knowledge profiles and the target question's profile. User knowledge profiles are derived according to Eq. (4).
KScore ranks experts by combining the *KProfile* and users' reputation scores based on Eq. (10). Users' reputation scores are calculated by Eqs. (11) and (12).
ExpertScore is a hybrid of *KScore* and *ExpertHITS* by Eq. (9).
QD-KProfile ranks experts according to the cosine similarities between users' knowledge profiles and the target question's profile. User knowledge profiles are derived according to Eq. (15), which considers the relevance (similarity) of the target question to the past questions answered by users.
QD-KScore ranks experts by combining the *QD-KProfile* and users' reputation scores based on Eq. (10). Users' reputation scores are calculated by Eqs. (16)–(18) considering the relevance of the target question and the past questions answered by users.
QD-ExpertScore is a hybrid of *QD-KScore* and *QD-HITS* by Eq. (9).

4.2. The approaches of extracting term vectors for deriving user knowledge profile

We compare four approaches to extract the term vectors of UQA pairs, which are derived from a given category/whole data set and with/without considering field weight. In addition, we compare two approaches to build up user knowledge profiles by aggregating UQA term vectors according to category or without considering the category. The experiment is conducted by using the analytical data that consists of 100 questions. Table 2 shows the result of recommending top-5 experts according to the similarity of user knowledge profiles and target question under various approaches. In deriving the term vectors, discriminating term weight (IDF) according to category range is better than taking the whole range into account. The approach considering the weight of field (question title, question description, and answer) is better than the one without considering field weight. When building up user knowledge profiles, the aggregation according to category (KP-category) is better than mixing all UQA together (KP-all). Note that the table lists the best result in bold values and the result of the approach chosen in underlined values.

We achieve better results in the experiment by deriving term weight (IDF) from category range. Moreover, the term should have different weights when it appears in different fields. The question title field has the smallest weight, while the answer field has the largest weight, since the answer is much longer and clear, and the terms in an answer are more representative. The result also shows that building up user knowledge profiles according to category is better. Users may have more than one knowledge domain. If we build up user knowledge profiles without considering category domain, the terms in different domains will be mixed together, and then the term vector cannot represent the subjects of user's knowledge domains appropriately. According to the result, for the rest of experiments, we derive the term vectors considering the

Table 2
Comparing different approaches in deriving term vectors.

IDF range	KP range					
	MRR		P@5		MAP	
	KP-all	KP-category	KP-all	KP-category	KP-all	KP-category
Whole range	0.702	0.735	0.470	0.476	0.388	0.382
Whole range with weight	0.731	0.768	0.482	0.514	0.401	0.410
Category range	0.723	0.761	0.474	0.494	0.387	0.397
Category range with weight	**0.764**	**0.790**	**0.502**	**0.524**	**0.412**	**0.424**

category range and field weight. Moreover, we use the category-based approach to aggregate UQA pairs in deriving user knowledge profiles.

4.3. The factors in deriving user knowledge profile

Besides the alternatives of TF-IDF and category, we also consider other factors when we build up user knowledge profiles. As mentioned in Section 3.3, there are time factor and quality measures (e.g. voting factor and evaluation factor) that influence the importance of each UQA. We use 100 analytical questions to conduct the experiment. Table 3 shows the experiment result of recommending top-5 experts based on the user knowledge profiles (KP) generated according to different factors, where "E" means evaluation factor, "T" means time factor, "V" means voting factor, and "none" means that none of the factor is used. The result shows that the evaluation factor does not contribute to improve the quality of finding experts. The result is even worse when we take evaluation factor into account. The reason may be that the evaluation is given for a QAA (question and all answers) not for a specific UQA (user-question–answer), and the evaluation given by users may be too general to differentiate the quality of UQA pairs. The best result is achieved when we consider the voting factor and time factor. The voting factor can discriminate the quality of a specific user's answer because it is given for a UQA pair. Moreover, the time factor is derived from the posting time of a question, i.e. the more recent UQA would be given a higher value. It is reasonable that the more recent a question is, the higher the probability that the answerer still has the up-to-date knowledge about the knowledge domain of the question. Moreover, the time factor can represent the activity of a user. If a user is active at recent time, he is more likely to answer questions. Accordingly, for the rest of our experiments, we use the voting factor and time factor to derive user knowledge profiles.

4.4. The effect of user reputation on deriving knowledge score

In this subsection, we decide the value of parameter λ in Eq. (11) for calculating user reputation, and the value of parameter α in Eq. (10) for deriving the knowledge score described in Section 3.4.1. The α is used to adjust the relative importance of user knowledge profile and user reputation, and we use λ to adjust the relative importance of adoption ratio and the number of best answers. The experiment is conducted by using the analytical dataset that consists of 100 questions. We first use NCR (normalized category reputation) as reputation score to compute the user knowledge scores and the result of recommending top-5 experts according to the knowledge scores is shown in Table 4.

The result shows that the performance rises as the value of α increases and the peaks of the performance are achieved at $\alpha = 0.9$ regardless of λ. Comparing different values of λ, the performance is best when λ is 0.5. This means that we should consider both the adoption ratio of best answers and the number of best answers, and the value of NCR should be half the value of the adoption ratio plus half the value of the adoption ratio multiplied by the number of best answers. The result shows that the relevance of user subject knowledge (knowledge profile) to the target question is more important than user reputation score in deriving user knowledge score. Accordingly, for the rest of our experiments, we use the NCR with $\lambda = 0.5$ to compute the reputation score, and we set $\alpha = 0.9$ in Eq. (10) to derive user knowledge score.

There are other factors that may be used for computing the reputation score. We try overall reputation *NR*, without considering category. Besides, the Yahoo! Answers website provides the function of adding friends with rating scores (e.g. 1–5) to a user's friend list. We derive the friend reputation (FR) of a user based on the weighted sum of the social rating scores s/he has received. The weight is determined by calculating the similarity between users according to their top five expert domains. Moreover, the Yahoo! Answers website assigns each user a knowledge grade that is computed from the user's answering records, such as the amount of answers and the ratio of adoption. There are seven knowledge grades defined. Except for the highest and lowest ones, every grade can be subdivided into five levels. There are 27 knowledge grade levels. We also try to use the translated knowledge grade (KG) as the reputation score.

Our result shows that NCR is better than NR, FR, and KG. NCR is calculated from the categorical domain rather than the whole domain as the NR is, and thus the NCR is more suitable than NR in finding experts for a question. The value of KG only has 27 kinds, and thus is not good enough to discriminate the different levels of user's reputations. The value of FR, which is calculated from social rating scores given by very few users, denotes a kind of social relationship and thus is not appropriate to denote a user's reputation or expertise on the subject domain.

Table 3
Comparing different factors in deriving knowledge profiles.

	MRR	P@5	MAP
KP-E	0.689	0.484	0.366
KP-ET	0.723	0.484	0.373
KP-none	0.712	0.492	0.376
KP-T	0.733	0.494	0.378
KP-VE	0.753	0.510	0.402
KP-V	0.773	0.516	0.416
KP-VET	**0.791**	0.522	0.419
KP-VT	0.790	**0.524**	**0.424**

Table 4
The performance of normalized category reputation (NCR) over α and λ.

α	MRR					P@5					MAP				
	λ														
	0	0.25	0.5	0.75	1	0	0.25	0.5	0.75	1	0	0.25	0.5	0.75	1
0	0.48	0.48	0.46	0.37	0.10	0.32	0.33	0.28	0.18	0.07	0.21	0.21	0.18	0.10	0.03
0.1	0.48	0.48	0.46	0.38	0.12	0.32	0.33	0.28	0.18	0.08	0.21	0.21	0.18	0.12	0.04
0.2	0.50	0.48	0.46	0.39	0.14	0.33	0.33	0.29	0.19	0.09	0.22	0.22	0.18	0.13	0.04
0.3	0.51	0.49	0.47	0.43	0.19	0.33	0.33	0.30	0.20	0.13	0.22	0.22	0.19	0.15	0.06
0.4	0.51	0.50	0.48	0.46	0.26	0.35	0.34	0.31	0.22	0.16	0.24	0.23	0.20	0.17	0.09
0.5	0.53	0.51	0.50	0.50	0.34	0.38	0.37	0.33	0.25	0.19	0.26	0.25	0.22	0.20	0.13
0.6	0.54	0.54	0.54	0.56	0.46	0.41	0.40	0.36	0.30	0.24	0.28	0.28	0.26	0.25	0.19
0.7	0.56	0.57	0.61	0.65	0.56	0.44	0.43	0.42	0.38	0.33	0.30	0.30	0.31	0.32	0.27
0.8	0.63	0.68	0.70	0.75	0.67	0.46	0.48	0.49	0.49	0.44	0.34	0.36	0.38	0.42	0.37
0.9	**0.79**	**0.82**	**0.85**	**0.84**	**0.82**	**0.53**	**0.53**	**0.56**	**0.55**	**0.52**	**0.43**	**0.45**	**0.48**	**0.48**	**0.44**
1	0.79	0.79	0.79	0.79	0.79	0.52	0.52	0.52	0.52	0.52	0.42	0.42	0.42	0.42	0.42

4.5. The relative importance of knowledge score and authority score

We used the HITS algorithm and PageRank algorithm to compute our authority score *A_score* and compare their performance. The experiment is conducted by using the analytical dataset that consists of 100 questions. Our preliminary result indicates that the HITS algorithm performs better than the PageRank algorithm when we only use the authority score to recommend top-5 experts ($\beta = 0$ in Eq. (9)). We also try different ways to calculate *A_score* using the HITS algorithm. The first one, HITS_B, uses best answerers to construct the asking-answering graph. The second one, HITS_H, takes 80% authority value and 20% hub value as the *A_score*. The last one, HITS, uses all the answerers to construct the asking-answering graph as described in Section 3.4.2. Note that the three approaches all use the respective questions and answers/best-answers in a specific category to construct the graph. Both the performances of HITS_H and HITS-B are very similar to that of the HITS. Thus, we use the authority value of HITS algorithm as our *A_score*.

After deciding the algorithm to calculate *A_score*, we try different β to compute the final expert score in Eq. (9). We also use 100 analytical questions to conduct the experiment. β is the parameter used to adjust the relative importance of knowledge score and authority score. Fig. 3 shows the result of recommending top-5 experts based on expert scores. For P@5 and MAP, the performance increases when β rises and peaks at $\beta = 0.8$, so the best performance is achieved by setting $\beta = 0.8$. The knowledge score is more important than the authority score to derive the final expert score.

4.6. Comparison of category-based methods with baseline methods

In previous sections, we used the analytical data to determine the factors and setting of the parameters in the proposed method. In this section, the performance of our category-based approaches is compared with that of other methods through the 100 testing questions. The baseline methods include *CBDM*, *VSM*, *KGrade*, *ExpertHITS* and *ExpertPRank*. We used the *CBDM* approach proposed by Liu et al. (2005) as one of the baseline methods. For a Cluster-based language model (*CBDM*), the expert profiles were built from past question–answer pairs (both question and answer texts) of a given category. For each candidate expert, we derived four category-based expert profiles, since there were 4 categories for the 100 testing questions. In each category, the 136 candidate experts were clustered into 5 clusters by means of the k-means algorithm. *CBDM* ranks experts according to the likelihood of the target question (query) being generated from each expert profile and cluster profile of a given testing question category. We used Bayesian smoothing with the Dirichlet prior technique to conduct the smoothing

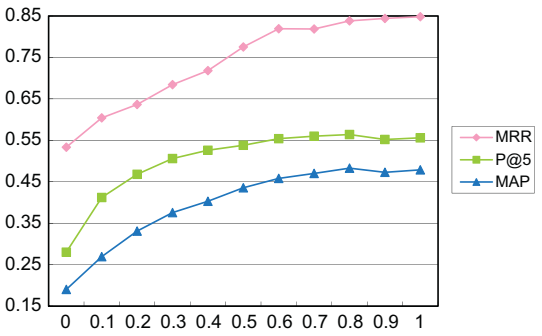

Fig. 3. The performance of category-based *ExpertScore* over β.

by varying different values of μ (the Dirichlet parameter). *VSM* ranks experts according to the cosine similarity of user profiles and the target question. The term vectors of UQA pairs are derived from a given category by considering the field weights of terms. After calculating the TFIDF, we built up user profiles by aggregating UQA term vectors of questions answered by users in a given category.

The *KGrade* method ranks experts by their "knowledge grade"; and if they have the same knowledge grade, the method ranks them by the number of questions that they had answered. The *ExpertPRank* ranks experts based on their PageRank scores as calculated by Eq. (14). *ExpertHITS* ranks experts based on their authority scores calculated by Eq. (13). There were variants of our approaches for comparison. The first one was *KProfile*, which was calculated using Eqs. (4) and (10) with $\alpha = 1$. The second was *KScore*, which was calculated using Eq. (10) with $\alpha = 0.9$ and $\lambda = 0.5$ in Eq. (11). The last was *ExpertScore*, which was a hybrid of *KScore* and *ExpertHITS* by applying Eq. (9) with $\beta = 0.8$, $\alpha = 0.9$ and $\lambda = 0.5$.

The result is shown in Table 5. The result shows that *CBDM* and *VSM* performed better than *ExpertHITS, ExpertPRank*, and *KGrade*. The baseline methods, including *VSM*, *KGrade*, *ExpertPRank* and *ExpertHITS*, performed worse than *KProfile*, *KScore* and *ExpertScore*. The link-analysis methods, including *ExpertPRank* and *ExpertHITS*, are designed to find experts for a given category, not for a target question. Thus, they may not be appropriate for finding the experts for a specific question. *KProfile*, *KScore* and *ExpertScore* are suitable for finding experts for a target question. Interestingly, *CBDM* performed better than *VSM* and *KProfile* for MRR, but worse than *VSM* and *KProfile* for p@5 and MAP. This implied that *CBDM* was more effective in finding the top-ranked experts, but its performance was worse than *VSM* and *KProfile* in finding a greater number of suitable experts. *KScore* and *ExpertScore* performed better than *CBDM* for all metrics.

KScore, which considers both user knowledge profiles and user reputation, performed better than *KProfile* which only considers user knowledge profiles. The user reputation score, which is derived from the adoption ratio and number of best answers, can ensure the quality of a user's answers, i.e. a user is more likely to be an expert when he has a higher user reputation score. Therefore, the method which considers both the relevance of the user's subject knowledge (profiles) to the target question and the quality measure (e.g. user reputation) performed better than the method with subject relevance only. From the result, we could also observe that the proposed *ExpertScore* that combined both *KScore* (considering user knowledge profile and user reputation) and *A_score* (computed by HITS algorithm), performed better than *KProfile* and *KScore*.

To examine the differences in performance between the proposed category-based methods and the baseline methods, we performed a statistical hypothesis test, the pair-wise and two-tailed *t*-test. Table 6 shows the *p*-values of the *t*-tests on MRR, P@5 and MAP for the comparison with the baseline methods. The results show that the differences were generally statistically significant at the 0.01 or 0.001 level, except for the comparison with *CBDM* on MRR. The MRR value of *CBDM* is higher than that of *KProfile*.

4.7. Comparison of question-dependent methods with other methods

We have also proposed question-dependent approaches, which build user knowledge profiles, user reputations, and weighted asking-answering graphs according to the relevance of past questions to the target questions, as described in Eqs. (15), (16), and (19) of Section 3.5. In this section, we have compared the question-dependent methods with the category-based methods (*KProfile*, *KScore*, *ExpertScore*). We also compared our methods with the baseline methods *CBDM*, *VSM*, *HITS* and *QD-HITS*. *QD-HITS* ranks experts based on their authority scores as calculated by Eq. (19), where the links of the asking-answering graph are weighted by the relevance of past questions to the target question.

The question-dependent methods included *QD-KProfile*, *QD-KScore*, and *QD-ExpertScore*. *QD-KProfile* was calculated using Eqs. (15) and (10) with $\alpha = 1$. *QD-KScore* was calculated using Eqs. (15)–(18), (10) with $\alpha = 0.9$ and $\lambda = 0.5$ in Eq. (16). *QD-ExpertScore* was a hybrid of *QD-KScore* and *QD-HITS* by applying Eq. (9). We first conducted an experimental analysis to decide the value of parameter β in deriving *QD-ExpertScore*. For the question-dependent approach, we used *QD-HITS* to compute the *A_score*, since the experiment result presented in Section 4.6 shows that the HITS algorithm performed better than the

Table 5
The performance of all approaches.

Methods	MRR	P@5	MAP
ExpertPRank	0.371	0.240	0.173
KGrade	0.427	0.265	0.155
ExpertHITS	0.575	0.322	0.244
VSM	0.630	0.412	0.295
CBDM	0.730	0.331	0.271
KProfile	0.710	0.445	0.336
KScore	0.787	0.505	0.406
ExpertScore	0.801	0.519	0.416
QD-HITS	0.576	0.340	0.253
QD-KProfile	0.721	0.455	0.344
QD-KScore	0.794	0.516	0.417
QD-ExpertScore	0.825	0.540	0.440

Table 6
The *t*-test results for the comparison of category-based approaches with baseline methods.

Methods	Baselines	MRR	*p*-Value	P@5	*p*-Value	MAP	*p*-Value
KProfile		0.710		0.445		0.336	
	ExpertHITS	0.575	0.0000***	0.322	0.0000***	0.244	0.0000***
	VSM	0.630	0.0000***	0.412	0.0016**	0.295	0.000126***
	CBDM	0.730	0.3991	0.331	0.0000***	0.271	0.0000***
KScore		0.787		0.505		0.406	
	ExpertHITS	0.575	0.0000***	0.322	0.0000***	0.244	0.0000***
	VSM	0.630	0.0000***	0.412	0.0000***	0.295	0.0000***
	CBDM	0.730	0.0070**	0.331	0.0000***	0.271	0.0000***
ExpertScore		0.801		0.519		0.416	
	HITS	0.575	0.0000***	0.322	0.0000***	0.244	0.0000***
	VSM	0.630	0.0000***	0.412	0.0000***	0.295	0.0000***
	CBDM	0.730	0.0011**	0.331	0.0000***	0.271	0.0000 ***

*Significance marked using $p < 0.05$.
** Significance marked using $p < 0.01$.
*** Significance marked using $p < 0.001$.

PageRank algorithm. The performance of *QD-ExpertScore* under different β is shown in Fig. 4. In general, the performance of the *QD-ExpertScore* method increased when β rose and peaked at $\beta = 0.8$, so we set $\beta = 0.8$ when using the question-dependent approach to compute *ExpertScore*.

Finally, we compared all the approaches through the 100 target (testing) questions, and the result is shown in Table 5. The *KGrade* method and the category-based link analysis methods, *ExpertPRank* and *ExpertHITS*, were still the worst approaches because they did not take the target questions into account. The question-dependent *HITS* (*QD-HITS*) method performed better than the category-based *ExpertHITS*, but generally performed worse than the *VSM* and *CBDM* methods. Our proposed methods generally outperformed the baseline methods. *QD-KProfile* performed better than *CBDM* for p@5 and MAP, but worse than *CBDM* for MRR. *QD-KScore* and *QD-ExpertScore* performed better than *CBDM* for all metrics. Table 7 shows the *p*-values of the pair-wise and two-tailed *t*-tests on MRR, P@5 and MAP for the comparison of question-dependent approaches with the baseline methods. The results show that the differences were generally statistically significant at the 0.01 or 0.001 level, except for the comparison with *CBDM* on MRR. The MRR value of *CBDM* is higher than that of *QD-KProfile*.

Moreover, the comparison of category-based methods with question-dependent methods is as follows. The *QD-KScore* method that considers user reputation performed better than the *QD-KProfile* method. The *QD-ExpertScore* method, which further considers the authority score, performed better than the *QD-KScore* method that does not consider the authority score. Moreover, *QD-KProfile* performed better than *K-Profile*; *QD-KScore* performed better than *KScore*; and *QD-ExpertScore* performed better than *ExpertScore*. The question-dependent approaches which considered the relevance of past questions to the target question performed better than the category-based approaches which did not. The result shows that the question-dependent method *QD-ExpertScore* performed the best when finding appropriate experts to answer the target question.

4.8. Significance tests for verifying various effects

We conducted the pair-wise and two-tailed *t*-tests to verify the effect of considering user reputation and authority score on the quality of expert finding, as shown in Table 8. The differences of *KScore* vs. *KProfile* and *QD-KScore* vs. *QD-KProfile* were statistically significant at the 0.001 level. Thus, considering user reputation definitely contributed to improving the quality of expert finding. The difference of *ExpertScore* vs. *KScore* was not significant for the *t*-tests on MRR, p@5 and MAP. The difference of *QD-ExpertScore* vs. *QD-KScore* was statistically significant at the 0.01 or 0.001 level. This implied that considering the authority score also contributed to improving the quality of the hybrid *QD-ExpertScore* approach for expert finding. However, the enhancement was not significant for the category-based *ExpertScore* method.

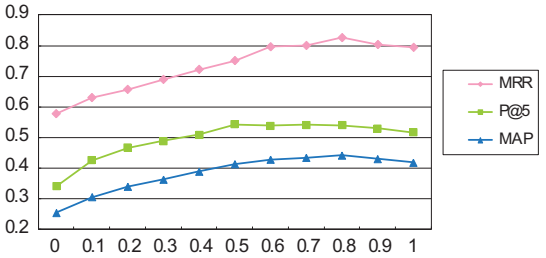

Fig. 4. The performance of question-dependent *QD-ExpertScore* over β.

Table 7
The *t*-test results for the comparison of question-dependent approaches with baseline methods.

Methods	Baselines	MRR	*p*-Value	P@5	*p*-Value	MAP	*p*-Value
QD-KProfile		0.721		0.455		0.344	
	QD-HITS	0.576	0.0000***	0.340	0.0000***	0.253	0.0000***
	VSM	0.630	0.0000***	0.412	0.0000***	0.295	0.0000***
	CBDM	0.730	0.7264	0.331	0.0000***	0.271	0.0000 ***
QD-KScore		0.794		0.516		0.417	
	QD-HITS	0.576	0.0000***	0.340	0.0000***	0.253	0.0000***
	VSM	0.630	0.0000***	0.412	0.0000***	0.295	0.0000***
	CBDM	0.730	0.0038**	0.331	0.0000***	0.271	0.0000 ***
QD-ExpertScore		0.825		0.540		0.440	
	QD-HITS	0.576	0.0000***	0.340	0.0000***	0.253	0.0000***
	VSM	0.630	0.0000***	0.412	0.0000***	0.295	0.0000***
	CBDM	0.730	0.0000***	0.331	0.0000***	0.271	0.0000 ***

*Significance marked using $p < 0.05$.
** Significance marked using $p < 0.01$.
*** Significance marked using $p < 0.001$.

Table 8
The *t*-test results for verifying various effects.

Effects	Methods	MRR	*p*-Value	P@5	*p*-Value	MAP	*p*-Value
Effect of reputation	*KScore*	0.787	0.0000***	0.505	0.0000***	0.406	0.0000***
	KProfile	0.710		0.445		0.336	
	QD-KScore	0.794	0.0000***	0.516	0.0000***	0.417	0.0000***
	QD-KProfile	0.721		0.455		0.344	
Effect of authority	*ExpertScore*	0.801	0.1121	0.519	0.0521	0.416	0.0640
	KScore	0.787		0.505		0.406	
	QD-ExpertScore	0.825	0.0000***	0.540	0.0014**	0.440	0.0002***
	QD-KScore	0.794		0.516		0.417	
Effect of question-dependency	*QD-KProfile*	0.721	0.1159	0.455	0.0017**	0.344	0.0027**
	KProfile	0.710		0.445		0.336	
	QD-KScore	0.794	0.4513	0.516	0.0526	0.417	0.0215*
	KScore	0.787		0.505		0.406	
	QD-ExpertScore	0.825	0.0014**	0.540	0.0002***	0.440	0.0000***
	ExpertScore	0.801		0.519		0.416	

* Significance marked using $p < 0.05$.
** Significance marked using $p < 0.01$.
*** Significance marked using $p < 0.001$.

Table 9
The performance of all approaches based on the best answers of testing questions.

Methods	MRR	Precision	MAP
ExpertPRank	0.1037	0.034	0.1037
KGrade	0.1398	0.052	0.1398
ExpertHITS	0.1352	0.056	0.1352
VSM	0.3630	0.140	0.3630
CBDM	0.4798	0.132	0.4798
Kprofile	0.4527	0.152	0.4527
Kscore	0.4707	0.134	0.4707
ExpertScore	0.4777	0.144	0.4777
QD-HITS	0.1438	0.06	0.1438
QD-Kprofile	0.4872	0.162	0.4872
QD-Kscore	0.5085	0.142	0.5085
QD-ExpertScore	0.5273	0.152	0.5273

Finally, we conducted the *t*-test to verify the effect of the question-dependent approach on the quality of expert finding. Table 8 also shows the *p*-values of the *t*-tests on MRR, P@5 and MAP for the comparison of each pair of question-dependent approach vs. category-based approach. The result shows that the differences were statistically significant at the 0.05 level for the *t*-tests on MAP. Particularly, the differences of *QD-ExpertScore* vs. *ExpertScore* were statistically significant at the 0.01 or

0.001 level for the *t*-tests on all measures. However, the differences of *QD-KProfile* vs. *KProfile* and *QD-KScore* vs. *KScore* were not significant for the *t*-test on MRR. The difference of *QD-KProfile* vs. *KProfile* was statistically significant at the 0.01 level for the *t*-tests on P@5. However, the difference of *QD-KScore* vs. *KScore* was not significant for the *t*-test on P@5. The question-dependent approaches performed better than the category-based approaches because they considered the relevance of the target question and the past questions, although the improvement on MRR may not be significant. Considering the relevance of past questions to the target question led to a better quality of expert finding.

4.9. Comparison of various methods based on the best answers of testing questions

We have also compared all approaches, by using the user who gave the best answer of a testing question as the suitable expert of the question, instead of using the experts decided by the human raters. The results are shown in Table 9. In general, the performance trends of various methods are similar to the results evaluated by using the experts decided by the human raters. Our proposed category-based methods generally performed better than the baseline methods, except for the *CBDM* on MRR (MAP). The precision value of *KProfile/KScore/ExpertScore* is higher than that of *CBDM*, while the MRR (MAP) value of *CBDM* is higher than that of *KProfile/KScore/ExpertScore*. The proposed question-dependent approaches performed better than all baseline methods. The question-dependent approaches generally performed better than the category-based approaches. The result also shows that the question-dependent method *QD-ExpertScore* performed the best when finding appropriate experts to answer the target questions.

5. Conclusion and future work

In this paper, we propose an effective hybrid approach to find appropriate experts to answer target questions and to promote the process of knowledge sharing through solving the problem of waiting for qualified experts to answer the question. Our contribution comprises addressing the issue of finding experts for target questions by considering the user subject relevance of user knowledge profile to the target question, user reputation, and user authority. We conduct experiments using a data set obtained from Yahoo! Answers in Taiwan to compare our methods with other conventional methods. We have tried several approaches to generate user knowledge profiles: (1) derived term vectors from a given category/whole data set and with/without considering field (question title/description and answer) weight as weight of term, (2) considered a question–answer's quality measures (voting factor and evaluation factor) and time factor as the weight to generate a given user's knowledge profile from aggregating the profile of his question–answer pairs, and (3) took the category of question–answer pair into account to make the result of finding experts more effective and accurate.

Our experiment result in deriving user knowledge profiles shows that using the category dataset for deriving IDF (inverse documents frequency) is better than using the whole dataset. Considering the field weight of term is better than without considering the field weight; and building knowledge profiles according to category is better than according to the whole data set. Moreover, considering the quality measure (e.g. voting factor) and the time factor as the weight of UQA can improve the quality of computing user knowledge profiles.

Our result also implies that by considering user reputation, we can improve the quality of deriving knowledge score for expert finding, while the subject relevance of user knowledge profiles to the target question is more important than user reputation score for expert finding. The proposed *ExpertScore* method, which combines the knowledge score and authority score, performs better than other conventional methods such as *ExpertHITS*, *ExpertPRank*, *VSM* and *CBDM*. In addition, setting $\beta = 0.8$ in Eq. (9) for deriving the *ExpertScore* can result in best performance. The knowledge score considering the subject relevance of user knowledge profile and user reputation contributes more in expert finding than the authority score derived from the link analysis.

Furthermore, our proposed category-based approaches can be extended to develop question-dependent approaches, i.e. building up user knowledge profiles, computing user reputations and user authority by considering the relevance of past questions to the target question, which in turn leads to a better result of finding experts for a target question. We use the similarity of the target question and past questions as the weight when calculating user knowledge profile, user reputation and user authority. The experimental result shows that the extended question-dependent approach, *QD-ExpertScore*, performs best when $\beta = 0.8$, and it is the best one among all methods.

We conducted *t*-tests to examine the differences in performance between the proposed methods and the baseline methods. Our proposed methods generally performed better than the baseline methods with statistically significant differences, except that *CBDM* performed better than *KProfile* and *QD-KProfile* for MRR. The *t*-test results also show that the differences in performance improvement considering user reputation and authority score were generally statistically significant. However, the enhancement based on user authority was not significant for the category-based *ExpertScore* method. The question-dependent approaches generally performed better than the category-based approaches with statistically significant differences, although the differences on MRR were not significant for *QD-KProfile* and *QD-KScore*.

There are some limitations to our experiment. There is no standard solution indicating who is an appropriate expert to answer the target question. Thus, human evaluation is conducted by asking human raters to justify the suitable experts for the target question. Human evaluation may have biases because human raters may consider different criteria to evaluate experts. In addition, the content of the question–answer pairs is colloquial and miscellaneous, so it is hard to extract the

words that can represent the questions/answers accurately. Synonyms or short forms of names also cause problems of extracting words because they are regarded as different "terms". To solve those problems, we plan to construct the ontology and domain dictionary for enhancing the accuracy of extracting term vectors in the future. Furthermore, our approach linearly combines the textual feature-based profiling method and the non-textual based link analysis method. We will try different approaches to combine them in our future work. Finally, we will extend our approach to find best answers for target questions in order to further recommend high quality answers as 'best answers', and advance the quality of knowledge sharing.

Acknowledgement

This research was supported in part by the National Science Council of the Taiwan under the Grant NSC 99-2410-H-009-034-MY3 and NSC 100-2410-H-009-016.

References

Balog, K., Azzopardi, L., & de Rijke, M. (2009). A language modeling framework for expert finding. *Information Processing and Management, 45*(1), 1–19.
Balog, K., & de Rijke, M. (2007). Determining expert profiles with an application to expert finding. In *IJCAI '07: Proceedings of the 20th intl joint conf. on artificial intelligence* (pp. 2657–2662).
Bian, J., Liu, Y., Agichtein, E., & Zha, H. (2008). Finding the right facts in the crowd: Factoid question answering over social media. In *WWW 2008: Proceedings of the 17th international world wide web conference* (pp. 467–476). Beijing, China: ACM Press.
Blooma, M. J., Chua, A. Y. K., & Goh, D. H.-L. (2008). A predictive framework for retrieving the best answer. In *SAC'08: Proceedings of the 23rd ACM symposium on applied computing* (pp. 1107–1111). Fortaleza, Ceara, Brazil: ACM Press.
Bouguessa, M., Dumoulin, B., & Wang, S. (2008). Identifying authoritative actors in question-answering forums – The case of Yahoo! Answers. In *KDD'08: Proceedings of the 14th ACM international conference on knowledge discovery & data mining* (pp. 866–874). Las Vegas, Nevada, USA: ACM Press.
Chen, W., Zeng, Q., & Liu, W. (2006). A user reputation model for a user-interactive question answering system. *Concurrency and Computation: Practice and Experience, 19*(5), 2091–2103.
Demartini, G., Gaugaz, J., & Nejdl, W. (2009). A vector space model for ranking entities and its application to expert search. In *Proceedings of the 31st European conference on information retrieval (ECIR 2009). LNCS* 5478 (pp. 189–201).
Fang, H., & Zhai, C. (2007). Probabilistic models for expert finding. In *Proceedings of the 29th European conference on information retrieval (ECIR 2007). LNCS* 4425 (pp. 418–430).
Jeon, J., Croft, W. B., Lee, J. H., & Park, S. (2006). A framework to predict the quality of answers with non-textual features. In *SIGIR'06: Proceedings of the 29th annual international ACM SIGIR conference* (pp. 228–235). Seattle Washington, USA.
Jurczyk, P., & Agichtein, E. (2007). Discovering authorities in question answer communities by using link analysis. In *CIKM'07: Proceedings of the 16the international conference on information and knowledge management*. Lisboa, Portugal: ACM Press.
Kim, S., Oh, J. S., & Oh, S. (2007). Best-answer selection criteria in a social Q&A site from the user-oriented relevance perspective. In *Proceedings of the 70th annual meeting of the American society for information science and technology*.
Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM, 46*(5), 604–632.
Liu, X., Croft, W. B. (2004). Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 186–193).
Liu, X., Croft, W. B., & Koll, M. (2005). Finding experts in community-based question answering services. In *CIKM'05: Proceedings of the 14the international conference on information and knowledge management*. Bremen, Germany: ACM Press.
Macdonald, C., Hannah, D., & Ounis, I. (2008). High quality expertise evidence for expert search. In *Proceedings of the 30th European conf. on information retrieval (ECIR 2008). LNCS* 4956 (pp. 283–295).
Macdonald, C., & Ounis, I. (2008). Voting techniques for expert search. *Knowledge and Information System, 16*, 259–280.
Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
Page, L., Brin, S., Motwani, R., & Winograd., T. (1998). The pagerank citation ranking: Bringing order to the Web. *Stanford digital library technologies project*.
Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management, 24*(5), 513–523.
Serdyukov, P., & Hiemstra, D. (2008). Modeling documents as mixtures of persons for expert finding. In *Proceedings of the 30th european conference on information retrieval (ECIR 2008), LNCS* 4956 (pp. 309–320).
Serdyukov, P., Rode, H., & Hiemstra, D. (2008). Modeling multi-step relevance propagation for expert finding. In *CIKM'08: Proceedings of the 17the international conference on information and knowledge management*. ACM Press.
Suryanto, M.A., Lim, E.-P., Sun, A., & Chiang, R. H. L. (2009). Quality-aware collaborative question answering: methods and evaluation. In *WSDM'09: Proceedings of the second ACM international conference on web search and data mining*.
Tho, Q. T., Hui, S. C., & Fong, A. C. M. (2007). A citation-based document retrieval system for finding research expertise. *Information Processing and Management, 43*(1), 248–264.
Voorhees, E. M. (1999). TREC-8 question answering track report. In *Proceedings of the 8th text retrieval conference*.
Zhang, J., Ackerman, M. S., & Adamic, L. (2007). Expertise networks in online communities: Structure and algorithms. In *WWW 2007: Proceedings of the16th international world wide web conference* (pp. 221–230).
Zhang, J., Ackerman, M.S., Adamic, L., & Nam, K.K. (2007). QuME: A mechanism to support expertise finding in online help-seeking communities. In *UIST'07: Proceedings of the 20th ACM symposium on user interface software and technology* (pp. 111–114).