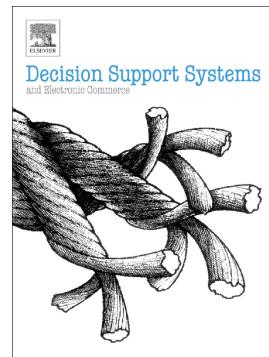


# Journal Pre-proof

Preference Enhanced Hybrid Expertise Retrieval System in Community Question Answering Services

Dipankar Kundu, Rajat Kumar Pal, Deba Prasad Mandal



PII: S0167-9236(19)30193-9

DOI: <https://doi.org/10.1016/j.dss.2019.113164>

Reference: DECSUP 113164

To appear in: *Decision Support Systems*

Received date: 17 April 2019

Revised date: 24 September 2019

Accepted date: 26 September 2019

Please cite this article as: D. Kundu, R. Kumar Pal and D. Prasad Mandal, Preference Enhanced Hybrid Expertise Retrieval System in Community Question Answering Services, *Decision Support Systems* (2019), <https://doi.org/10.1016/j.dss.2019.113164>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

# Preference Enhanced Hybrid Expertise Retrieval System in Community Question Answering Services

Dipankar Kundu<sup>a,\*</sup>, Rajat Kumar Pal<sup>a</sup>, Deba Prasad Mandal<sup>b</sup>

<sup>a</sup>*Department of Computer Science and Engineering, University of Calcutta,  
Kolkata-700073, India*

<sup>b</sup>*Machine Intelligence Unit, Indian Statistical Institute, Kolkata-700108, India*

## Abstract

Here, we propose a preference enhanced hybrid expertise retrieval (PEHER) system in community question answering services. PEHER consists of three segments, namely, preferability estimator, authority estimator, and expertise estimator. The preferability estimator utilizes the textual information to determine both intra-profile and inter-profile preferences of answerers for each term. The intra-profile preferences consider the preference of a term using the answering history of a given answerer. The inter-profile preferences incorporate the preferences of all answerers for a term. These preferences are then used to determine the preferability of each answerer for each of the archived questions. The authority estimator considers the textual familiarity between each archived question and the profile of each answerer as the weight of the associated link in the network. The expertise estimator is composed of three blocks, namely, question similarity finder, proficiency estimator, and expert list generator. The question similarity finder finds the similarities

---

\* Corresponding author. Email addresses: d.kundu7681@gmail.com (Dipankar Kundu), pal.rajatk@gmail.com (Rajat Kumar Pal), dpmandal@isical.ac.in (Deba Prasad Mandal)

between the new question and each of the archived questions. The proficiency estimator uses the said similarities of the archived questions along with their preferabilities to decide the proficiencies of answerers for the new question. Finally, the expert list generator considers the authorities and proficiencies to generate a list of experts for a given question. We compare PEHER with twenty existing methods on four real-world datasets using five performance measures. We find that PEHER outperforms the comparing algorithms in 92.00% (368 out of 400) cases.

*Keywords:* community question answering, expertise retrieval, hybrid systems, social network analysis.

## 1. Introduction

Community question answering (CQA) services allow a user to post her doubts in the form of questions and other users to answer those questions. CQA portals, such as Yahoo!Answers<sup>1</sup>, Quora<sup>2</sup>, Stack Overflow<sup>3</sup>, and Wikianwers<sup>4</sup>, are popular. Although such services are quite helpful for information seekers, the existence of a substantial amount of unanswered questions is a problematic issue in CQA services. To address this issue, researchers have proposed several approaches, which include the concept of *question routing*. When a new question arrives, question routing is the process of forwarding the question to its appropriate experts. *Expertise retrieval*, the process of finding experts for a given question is an essential part of question routing.

There exist several types of approaches to expertise retrieval in CQA ser-

---

<sup>1</sup> <http://answers.yahoo.com/>    <sup>2</sup> <http://www.quora.com/>    <sup>3</sup> <http://stackoverflow.com/>

<sup>4</sup> <http://answers.wikia.com/>

vices including text-based [1–5], network-based [6–13], and hybrid approaches [14–19]. Usually, text-based approaches [1–5] utilize the textual information to reflect a specific and detailed aspect of an answerer’s expertise for a question. These approaches usually do not consider the preference of an answerer regarding her practice of terms. Conversely, network-based approaches [6–13] usually consider the relationships among the participants to determine the authority of an answerer. Mostly, network-based approaches overlook the textual information available in CQA services. Hybrid approaches [14–19] that combine both text-based and network-based approaches overcome some of their limitations and amalgamate their advantages.

Here, we propose a preference enhanced hybrid expertise retrieval (PEHER) system in CQA services that uses both textual information and a network. PEHER is composed of three segments: preferability estimator, authority estimator, and expertise estimator. The preferability estimator incorporates the preference of an answerer regarding terms. For this, we introduce two types of preferences: intra-profile preferences and inter-profile preferences. Intra-profile preferences consider the preference for a given term based on the performance profile of a given answerer. Again, for a given term, inter-profile preferences consider the preferences of all answerers. We incorporate these two preferences with the importance of terms to determine the preferability of an answerer for each archived question. The authority estimator uses a competition based expertise network (CBEN) [12] and incorporates textual information to determine the authority of a given answerer. Here, we assume that an answerer is familiar to a given question if she has answered a large number of similar questions. For this, we propose a measure

of familiarity which is realized as the link weights of the CBEN. The expertise estimator, again, has three blocks, namely, question similarity finder, proficiency estimator, and expert list generator. The question similarity finder computes the similarities between the archived questions and the new question. To determine the proficiency of the answerer, the proficiency estimator uses (i) the preferabilities of an answerer for the archived questions and (ii) the similarities of the archived questions with the new question. Finally, the expert list generator uses a new fusion scheme to combine the authorities and proficiencies to generate a group of experts. We use 4 real-world datasets to evaluate the comparative performance of the proposed system with 20 state-of-the-art methods using 5 measures. We find that in 92.00% (368 out of 400) cases, the proposed system outperforms the comparing algorithms.

## 2. A Brief Literature Review

There are several notable approaches to expertise retrieval in CQA services. These approaches can be categorized using many possible ways. However, here we identify three prominent categories: (i) text-based [1–5], (ii) network-based [6–13], and (iii) hybrid [14–19] approaches. Note that, there exist a few methods [20–22] that do not fall into these three categories. In this literature review, we restrict ourselves to only some of the most relevant and important works on expertise retrieval in CQA services.

### 2.1. *Text-based Expertise Retrieval*

One of the first prominent work on text-based expertise retrieval in CQA service is in [1]. It realizes expertise retrieval as an information retrieval problem. To be more specific, in that work, the authors used the textual

information associated with answerers in three different language models: (i) cluster based language model [23], (ii) query likelihood model [24], and (iii) relevance model [25]. In [2], researchers used the textual information associated with answerers to design a generalized topic model for expert finding. Later, researchers in [3] proposed a question routing method to rank answerers for a question. This method estimates the expertise of an answerer by taking into account three issues: (i) the textual similarity between the profile of the answerer with the new question, (ii) the availability of the answerer, and (iii) the quality of answers provided by the answerer. The authors treated the prediction of the availability of an answerer as a time-series problem. An extension of the work [3] in [4] also incorporates question category information to estimate the expertise of an answerer. In [5], the authors proposed a text-based method for expert finding using a concept of *theme* in the query likelihood language (QLL) model [26]. The *theme* of a query was determined based on the parts-of-speech of the words in a question.

## 2.2. Network-based Expertise Retrieval

One of the initial notable works on network-based approaches is the community expertise network (CEN) [6]. In this approach, the authors proposed two measures of authority estimation: ExpertiseRank and z-score. To estimate the authority, ExpertiseRank uses PageRank algorithm [27], whereas, z-score considers the in-degrees and out-degrees of the network. The works in [7] and [8] first construct CEN, and then, apply Hyperlink-Induced Topic Search (HITS) algorithm [28] to estimate the authority of an answerer in the community. Again, an extended category link graph [9, 10] approach measures the relevances among categories and constructs an extended network to

estimate the authority of an answerer. Another network-based method, Personalized PageRank [11], uses a personalization vector to assign preferences of answerers to estimate their authorities. A competition based expertise network (CBEN) for expert finding was proposed in [12]. This method considers the inherent competition among the co-answerers to select the best answerer and builds a CBEN. Then, it uses a link analysis technique in the network for finding authoritative answerers in the community. A community-aware network approach [13] estimates the community-aware authority of an answerer. First, it detects a set of overlapping communities, and then, it calculates the authority of an answerer.

### *2.3. Hybrid Expertise Retrieval*

There are at least three straightforward ways of hybridization of text-based and network-based expertise retrievals. They are : (i) to integrate the results obtained from different approaches; (ii) to use the textual similarity in the computation associated with the network; and (iii) to incorporate the textual information to develop a topic model, and then, to use the information from the topic model to construct a network. In the present investigation, we make use of the first two strategies, and hence, we restrict here our discussion on these two strategies.

The latent link analysis method [14] incorporates textual information to construct a network. This method uses the co-occurrence model [29] and calculates preliminary expert scores using textual similarity. Then, it considers a few experts with the top scorers to build a network. It computes the final authority scores by applying the propagation-based link analysis technique proposed in [30] on the constructed network.

The hybrid expertise retrieval method in [15] integrates a text-based method and a network-based method. Its text-based method incorporates (i) the best answer ratio of each answerer and (ii) the textual similarities between the new question and the profile of the answerer. The network-based method, again, constructs a CEN [6] using the relationship between the asker and the answerer, and then, applies a link analysis technique to estimate the authority of an answerer. The outcomes of these two methods are finally fused using a linear data fusion technique. Later, the authors in [16] extended the work in [15] by incorporating the concept of the relevance of a new question. This extended work used the textual similarity between a new question and each archived question to measure the relevance.

“ExpertRank” [17], combines a document-based relevance model and a network-based authority model. Again, the method in [18] considers a profile-based method for finding potential answerers of given a question. Moreover, it captures topic interests, authorities, and activity levels of the answerers. The algorithm then combines the scores of these approaches to determine the final expert score.

In [19], we proposed a hybrid expertise retrieval system in CQA services. It integrates the outcomes of a text-based segment, called knowledge analyzer, and a network-based segment, called authority analyzer. In that work, we proposed two variants of the method: one of them uses PageRank and the other one uses HITS. Moreover, we used a reciprocal rank fusion (RRF) [31] technique to use the knowledge and authority of each user.

### 3. The Objective, the Philosophies, and the Contributions of the Present Investigation

We introduce the following notations first, which we used throughout this article:

1. A question  $q = \{t_1, t_2, \dots, t_{|q|}\}$  is the set of unique terms present in the text associated with the question. Moreover,  $n(t_i, q), i \in \{1, 2, \dots, |q|\}$ , is the frequency of the term  $t_i$  in  $q$ .
2.  $Q = \{q_1, q_2, \dots, q_{|Q|}\}$  is the set of all questions. Here,  $|\{\cdot\}|$  indicates the cardinality of the set  $\{\cdot\}$ .
3.  $A = \{a_1, a_2, \dots, a_{|A|}\}$  is the set of all answerers.
4.  $A_q \subseteq A$  is the set of all answerers who answered the question  $q \in Q$ .
5.  $Q_a \subseteq Q$  is the set of all questions answered by the answerer  $a \in A$ . In this work, we consider  $Q_a$  as the performance profile of  $a$ .
6.  $T$  is the set of all unique terms present in  $Q$ .
7.  $T_a \subseteq T$  is the set of all unique terms present in  $Q_a \subseteq Q$ .
8.  $\text{user\_freq}[t] = |\{\cup_{a \in A, t \in T_a} a\}|$  is the number of answerers who have answered at least one question containing the term  $t \in T$ .
9.  $\text{term\_freq}[a, t]$  is the number of questions answered by  $a \in A$  with the term  $t \in T_a$ .

The objective of this work is to predict the answerers having expertise on a new question  $\hat{q} \notin Q$ . For  $\hat{q}$ , the system predicts a score  $\mathcal{E}(a_j, \hat{q})$  for each answerer  $a_j \in A$ . The system chooses the top-scoring subset of answerers  $A_{\hat{q}}^* \subseteq A$  as the possible experts on  $\hat{q}$ .

In the present investigation, we consider the following philosophies/assumptions.

1. A high value of  $\text{term\_freq}[a, t]$  indicates a high preference of the answerer  $a$  for the term  $t$ . In other words, answerer  $a$  prefers a term  $t_1 \in T_a$  more than another term  $t_2 \in T_a$ , if  $\text{term\_freq}[a, t_1] > \text{term\_freq}[a, t_2]$ .
2. Let  $T_{a_1a_2} = T_{a_1} \cap T_{a_2}$  be the common terms associated with two answerers  $a_1, a_2 \in A$ . Then, the preference of  $a_1$  for a term  $t \in T_{a_1a_2}$  is higher than the preference of  $a_2$  for  $t$ , if  $\text{term\_freq}[a_1, t] > \text{term\_freq}[a_2, t]$ .
3. According to the assumption of inverse document frequency (IDF)[32], the frequently used terms are common, and hence, are less discriminative. Following this, we assume that each term  $t \in T$  is not equally important. If  $t$  is used by a smaller number of answerers, it is more important, i.e., the importance of  $t$  is inversely proportional to  $\text{user\_freq}[t]$ .

Here, we define the importance of  $t$ ,  $\text{imp}[t]$ , as follows:

$$\text{imp}[t] = \log(|A| / \text{user\_freq}[t]). \quad (1)$$

4. The best answerer of a question deserves a higher credit when other answerers with high expertise have answered the same question. To realize it, let us assume that for a given question  $q \in Q$ ,  $\exists a_1, a_2 \in A_q \subseteq A$  and  $a_1$  is the best answerer. Then, the credit that  $a_1$  deserves should increase with an increase in the expertise level of  $a_2$ . Here, we assume that if  $a_2$  has answered a large number of questions that are similar (familiar) to  $q$ , then  $a_2$  has high expertise on  $q$ .
5. The performance of a hybrid expertise retrieval system depends on the information fusion techniques associated with it. Therefore, for new components of an expertise retrieval system, it is important to find a suitable information fusion technique.

Incorporating the aforementioned philosophies/assumptions in this work, we contribute in the following ways. First, we propose a score to measure the preferences of the answerers using their answering histories. Second, we propose a text-based approach to estimate the proficiency of each answerer by incorporating the preferences of the answerers. Third, we propose a modified CBEN to estimate the authority of each answerer in the community. The modified CBEN incorporates the familiarity of each archived question with each answerer. The said familiarity is estimated by calculating the similarity between each archived question and the performance profile of each answerer. Fourth, We introduce a new fusion strategy that effectively combines the outputs of the text-based and the network-based segments.

We now discuss the differences between the current work and our previous work in [19]. First, the text-based segment in [19] considers the hardness of a question and the question answerer association, whereas, here we employ the preferences of answerers in the proposed text-based segment. Second, the network-based segment in [19] incorporates answer quality, whereas, here the proposed network-based component takes into account the familiarities among the archived questions and the user profiles. Third, the work in [19] uses an existing fusion technique, called RRF [31]. On the contrary, we propose a new fusion scheme here.

#### **4. The Proposed Preference Enhanced Hybrid Expertise Retrieval (PEHER) System**

Figure 1a illustrates the proposed PEHER system. As shown in Fig 1a, PEHER takes the set of archived questions  $Q$ , the list of answerers  $A$ , and a

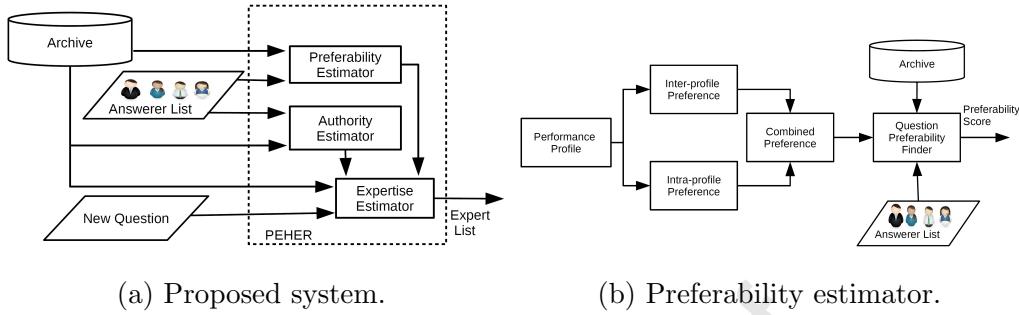


Figure 1: Architecture of the a) proposed system and b) preferability estimator

new question  $\hat{q}$  as inputs, and produces a list of experts as the output. Note that, PEHER is decomposed into three segments: (i) preferability estimator, (ii) authority estimator, and (iii) expertise estimator. The first two segments do not depend on the new question. Therefore, it is possible to perform computations a priori for these two segments. The last segment, expertise estimator, however, depends on the new question and hence, it needs to be processed online.

#### 4.1. Preferability Estimator

Figure 1b illustrates the architecture of the preferability estimator. We first compute both the intra-profile and inter-profile preferences of each answerer for each term. Then, we estimate a combined preference using these two scores. Next, we use this combined preference score into a block, called question preferability finder, to estimate a preferability score. Here, we would like to mention that the concept of using intra- and inter- profile preferences are inspired by similar definitions in the literature. For instance, in [33], the authors discussed the intra- and inter- consumer heterogeneity based methods for stated preference elicitation, especially for choice-based conjoint.

Moreover, in [34] a framework for estimating and updating user preferences is presented. The framework employs the Hierarchical Bayes estimator [33, 35], which deems for inter- and intra-consumer heterogeneity.

#### 4.1.1. Intra-profile Preference

We consider that an answerer  $a \in A$  prefers a term  $t_1 \in T_a$  compared to another term  $t_2 \in T_a$ , if  $\text{term\_freq}[a, t_1] > \text{term\_freq}[a, t_2]$ . Following this, we define the intra-profile preference  $\text{pref}[a, t]_{\text{IntraP}}$  of the term  $t$  by  $a$  as

$$\text{pref}[a, t]_{\text{IntraP}} = \text{term\_freq}[a, t] / \text{avg\_term\_freq}[a]. \quad (2)$$

Here,  $\text{avg\_term\_freq}[a]$  is the average frequency of each term used by the answerer  $a$ , which we estimate as

$$\text{avg\_term\_freq}[a] = \sum_{t \in T_a} \text{term\_freq}[a, t] / |T_a|. \quad (3)$$

Here,  $\sum_{t \in T_a} \text{term\_freq}[a, t]$  is the cumulative term frequency associated with all terms in  $T_a$ .

#### 4.1.2. Inter-profile Preference

Here, we assume that the preference of an answerer  $a_1 \in A$  for a term  $t \in T_{a_1 a_2} = T_{a_1} \cap T_{a_2}$  is higher than the preference of another answerer  $a_2 \in A$  for  $t$ , if  $\text{term\_freq}[a_1, t] > \text{term\_freq}[a_2, t]$ . Following this assumption, we assess the preference  $\text{pref}[a, t]_{\text{InterP}}$  of an answerer  $a$  for the term  $t$  as

$$\text{pref}[a, t]_{\text{InterP}} = \text{term\_freq}[a, t] / \text{avg\_freq}[t], \quad (4)$$

where  $\text{avg\_freq}[t]$  is the average term frequency of  $t$  defined as follows:

$$\text{avg\_freq}[t] = \sum_{a_1 \in \{\cup_{a_2 \in A, t \in T_{a_2}} a_2\}} \text{term\_freq}[a_1, t] / \text{user\_freq}[t]. \quad (5)$$

---

**Algorithm 1 : FUSE\_PREF( $\text{pref}[a, t]_{\text{IntraP}}, \text{pref}[a, t]_{\text{InterP}}$ )**


---

```

1: if  $\text{pref}[a, t]_{\text{IntraP}} \geq 1$  and  $\text{pref}[a, t]_{\text{InterP}} \geq 1$  then
2:    $\text{pref}[a, t]_c = \text{pref}[a, t]_{\text{IntraP}} + \text{pref}[a, t]_{\text{InterP}}$ 
3: else if  $\text{pref}[a, t]_{\text{IntraP}} < 1$  and  $\text{pref}[a, t]_{\text{InterP}} \geq 1$  then
4:    $\text{pref}[a, t]_c = \text{pref}[a, t]_{\text{InterP}}$ 
5: else
6:    $\text{pref}[a, t]_c = \text{pref}[a, t]_{\text{IntraP}} \times \text{pref}[a, t]_{\text{InterP}}$ 

```

---

Here,  $\{\cup_{a_2 \in A, t \in T_{a_2}} a_2\}$  is the set of all answerers who have used the term  $t$  and  $\sum_{a_1 \in \{\cup_{a_2 \in A, t \in T_{a_2}} a_2\}} \text{term\_freq}[a_1, t]$  is the cumulative frequency of  $t$ .

#### 4.1.3. Combined Preference

Next, we combine  $\text{pref}[\cdot, \cdot]_{\text{IntraP}}$  and  $\text{pref}[\cdot, \cdot]_{\text{InterP}}$ . For an answerer  $a$  and a term  $t$ , we denote the combined score  $\text{pref}[a, t]_c$ , which is computed following the procedure FUSE\_PREF as described in Algorithm 1.

We assume that a term  $t$  with a high  $\text{user\_freq}[t]$  is of low importance. In other words,  $t$  is more important if  $\text{user\_freq}[t]$  is low. With this assumption, we compute the preference of  $a$  for  $t$ , i.e.,  $\text{term\_pref}[a, t]$  as follows:

$$\text{term\_pref}[a, t] = \text{pref}[a, t]_c \times \text{imp}[t], \quad (6)$$

where  $\text{imp}[t]$  is as defined in (1).

#### 4.1.4. Question Preferability Finder

Using the preference of an answerer  $a$  for a term  $t$ , i.e.,  $\text{term\_pref}[a, t]$ , we estimate the preferability  $\text{q\_pref}[a, q]$  of  $a$  for a question  $q \in Q_a$  as

$$\text{q\_pref}[a, q] = \sum_{t \in q, q \in Q_a} \text{term\_pref}[a, t] / |q| \quad (7)$$

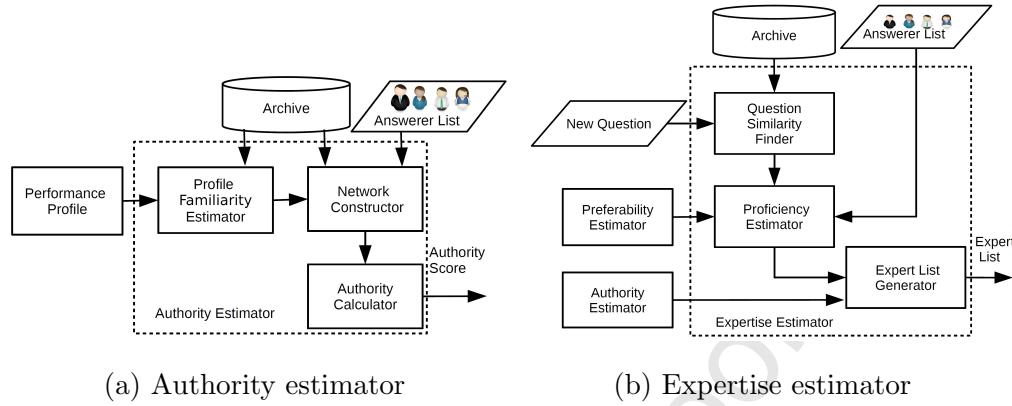


Figure 2: The architecture of the (a) authority estimator and (b) expertise estimator.

#### 4.2. Authority Estimator

Figure 2a visually demonstrates the workflow of the authority estimator segment. As Fig. 2a shows, we decompose this segment into three blocks, namely, profile familiarity estimator, network constructor, and authority calculator. The profile familiarity estimator computes the familiarity between each archived question and the performance profile of each answerer. Then, the network constructor uses these scores along with the inherent relationships among the answerers to construct a CBEN [12]. Next, the authority calculator uses a link analysis technique to estimate the authority of each answerer. Now, we discuss this process with further details.

##### 4.2.1. Profile Familiarity Estimator

For a question  $q \in Q$ , this block estimates the familiarity of an answerer  $a$  with  $q$ . If  $a$  has answered a large number of questions that are similar to  $q$ , then the familiarity,  $\text{fami}[a, q]$ , would be high. It is computed as

$$\text{fami}[a, q] = \mathcal{J}(a, q) / \sum_{\hat{a} \in A_q} \mathcal{J}(\hat{a}, q) \quad (8)$$

where  $\mathcal{J}(\cdot, \cdot)$  is the summation over the Jaccard similarity coefficients of  $q$  and the questions in  $Q_a$ . It is measured as follows.

$$\mathcal{J}(a, q) = \sum_{\dot{q} \in Q_a} (q \cap \dot{q}) / (q \cup \dot{q}). \quad (9)$$

#### 4.2.2. Network Constructor

The network constructor block constructs a CBEN [12]  $G = (A, E)$ , where  $A$  is the set of vertices (answerers) and  $E$  is the set of edges (relationships). Two answerers  $a_i, a_j \in A$  are connected using an edge ( $e_{ij} \in E$ ) in  $G$  if they have answered the same archived question, such that,  $a_j$  has been selected as the best answerer. Here, each edge  $e_{ij}$  is associated with a weight  $v_{ij}$ . Note that,  $e_{ij}$  and  $e_{ji}$  are two different links.

A CBEN [12] does not incorporate any textual information. However, we assume that incorporation of textual similarities between an archived question and the performance profiles of the answerers may enhance the performance. To be more specific, for a given question  $q \in Q$ , if an answerer  $a_1 \in A_q \subseteq A$  has defeated a co-answerer  $a_2 \in A_q \subseteq A$  with a high familiarity score, then  $a_1$  should be provided a higher credit. To incorporate this philosophy, we use  $\text{fami}[\cdot]$  in (8) as the weights of the edges as follows:

$$w_{ij} = \sum_{q \in Q} \text{fami}[a_i, q] \times \delta_{ij}^q; \quad (10)$$

$$\delta_{ij}^q = \begin{cases} 1 & \text{if } a_i, a_j \in A_q \text{ and } a_j \text{ is the best answerer,} \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

Moreover, we assume  $w_{ii} = 0$ ,  $\forall i$ , to avoid self-connection. The final weight

of the edge  $e_{ij}$  is then defined by normalizing the corresponding weight as:

$$v_{ij} = \begin{cases} w_{ij}/\sum_{m=1}^{|A|} w_{im} & \text{if } \sum_{m=1}^{|A|} v_{im} \neq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

#### 4.2.3. Authority Calculator

Following PageRank [27], we iteratively update the authority  $\text{auth}[a_j]$  of answerer  $a_j$  as

$$\text{auth}[a_j]_{(r+1)} = d \times \sum_{i=1}^{|A|} v_{ij} \times \text{auth}[a_i]_r + (1-d)/|A|, \quad (13)$$

until  $|\text{auth}[a_j]_{(r+1)} - \text{auth}[a_j]_r| \leq \tau$  or  $r = r_{max}$  hold. Here,  $d \in [0, 1]$  is the damping factor,  $r$  is the iteration index,  $\tau$  is a threshold, and  $r_{max}$  is the maximum number of iterations.

### 4.3. Expertise Estimator

Figure 2b shows the architecture of the expertise estimator segment. It is composed of three blocks, namely, question similarity finder, proficiency estimator, and expert list generator. Below, we discuss these with further details.

#### 4.3.1. Question Similarity Finder

In this block, we use the QLL model [26] to estimate the similarity between a new question  $\hat{q}$  and an archived question  $q \in Q$ . It is denoted by  $\text{sim}[\hat{q}, q]$  and is computed as

$$\text{sim}[\hat{q}, q] = \prod_{\hat{t} \in \hat{q}} \{\lambda p(\hat{t}|\theta_q) + (1-\lambda)p(\hat{t}|\theta_Q)\}^{n(\hat{t}, \hat{q})}. \quad (14)$$

Here,  $\theta_{(\cdot)}$  is the language model associated with  $(\cdot)$ ;  $\lambda \in [0, 1]$  is a user defined parameter; and  $n(\hat{t}, \hat{q})$  is as defined earlier.

#### 4.3.2. Proficiency Estimator

Now, we estimate the proficiency  $\text{profi}[a, \hat{q}]$  of an answerer  $a$  for a new question  $\hat{q}$  as

$$\text{profi}[a, \hat{q}] = \sum_{q \in Q_a} \text{sim}[\hat{q}, q] \times \text{q\_pref}[a, q]. \quad (15)$$

Here,  $\text{sim}[\hat{q}, q]$  and  $\text{q\_pref}[a, q]$  are as defined in (14) and (7), respectively.

#### 4.3.3. Expert List Generator

To generate the expert list, we fuse the proficiency ( $\text{profi}[a, \hat{q}]$ ) and the authority ( $\text{auth}[a]$ ) of an answerer  $a$  to estimate her expertise score ( $\mathcal{E}(a, \hat{q})$ ) for a new question  $\hat{q}$  using Algorithm 2. In Algorithm 2,  $\text{rank}[(\cdot)]$  indicates the rank using  $(\cdot)$ . The highest value is assigned to the rank 1 and the lowest value is assigned to the rank  $|A|$ . For instance, if there are four answerers  $a_1$ ,  $a_2$ ,  $a_3$ , and  $a_4$  with scores 0.5, 0.4, 0.4, and 0.3, respectively, then  $\text{rank}[a_1] = 1$ ,  $\text{rank}[a_2] = 2$ ,  $\text{rank}[a_3] = 2$ , and  $\text{rank}[a_4] = 4$ . Moreover,  $w_r$  is a balancing parameter that controls the effects of proficiency and authority when we use reciprocal rank fusion (RRF) [31] technique in Step 6 of Algorithm 2. Here, we assume that if an answerer has rank one with respect to either authority or proficiency, then she should be at the top of the list of experts. However, when an answer has rank one with respect to neither authority nor proficient, we use the RRF technique to determine her score.

## 5. Experiments, Results, and Discussions

### 5.1. Datasets and Measures

We use four real-world datasets: Movie, Music, Celebrity, and History [19]. We use five performance measures: mean reciprocal rank (MRR) [3],

---

**Algorithm 2 : FUSE\_RANK(profi[ $a, \hat{q}$ ], auth[ $a$ ])**

---

```

1:  $r_1 \leftarrow \text{rank}[\text{profi}[a, \hat{q}]]$ 
2:  $r_2 \leftarrow \text{rank}[\text{auth}[a]]$ 
3: if  $r_1 = 1$  or  $r_2 = 1$  then
4:    $\mathcal{E}(a, \hat{q}) = 1.0$ 
5: else
6:    $\mathcal{E}(a, \hat{q}) = \frac{w_r}{r_1} + \frac{1-w_r}{r_2}$ 

```

---

precision at the top N (P@N) [36], recall at the top N (R@N) [36], matching set count at the top N (MSC@N) [36], and accuracy [37].

### 5.2. The Cost of Implementation

We implement PEHER system in Java and execute it in a computer with an Intel i7 processor (3.60 GHz) and with 32 GB of RAM. The first two segments of PEHER system, i.e., the preferability estimator and the authority estimator can be executed in an off-line mode (as these two segments are independent of new questions). This reduces the real-time cost of PEHER significantly. Moreover, this offline part involves computation of well-known components, for instance, PageRank, CBEN, and Jaccard similarity, which are frequently used in a diverse set of popular algorithms. As off-line computation can be preprocessed, we emphasize on the real-time application cost. When a new question arrives, only the expertise estimator segment, needs to be executed. It involves, computation of equations (14), (15), and Algorithm 2. Now, the computation cost of (14), (15), and Algorithm 2 are  $O(|\hat{q}|)$ ,  $O(|Q_a|)$ , and  $O(1)$ , respectively. They are indeed low. However, they need to be executed for  $|Q|$ ,  $|A|$ , and  $|A|$  times, respectively. So, the cost of the application of PEHER increases with an increase in the number of archived questions and an increase in the number of answerers.

Table 1: List of Comparing Methods and Their References

#	Name of Comparing Method	#R	#	Name of Comparing Method	#R
1	CBEN-PageRank	[12]	11	Latent link analysis (LLA)	[14]
2	CBEN-HITS	[12]	12	NEWHITS without topic model	[41]
3	Document model (DM)	[38]	13	PageRank	[27]
4	Adaptive HITS	[39]	14	Personalized PageRank (P-PageRank)	[11]
5	Cluster-based language model (CBDM)	[1]	15	Query likelihood (QL) model	[1]
6	Community-aware PageRank based on speaker listener label propagation algorithm (CA-PR-SLPA)	[13]	16	TF-IDF	[42]
7	Community-aware HITS based on speaker listener label propagation algorithm (CA-HITS-SLPA)	[13]	17	TopicRank-based document priors (TBDP)	[43]
8	ExpertiseRank	[40]	18	Z-score	[40]
9	ExpertRank	[17]	19	Hybrid expertise retrieval system using PageRank (HER-PR)	[19]
10	HITS	[8]	20	Hybrid expertise retrieval system using HITS (HER-HITS)	[19]

### 5.3. Comparison With Other Methods

#### 5.3.1. Comparing Methods and Experimental Protocol

We compare PEHER with the 20 existing methods listed in Table 1. We borrow their results from [19]. The parameter settings for all these comparing methods are available in Sections 4.1.2 and 4.1.3 of [19]. PEHER requires six parameters. Table 2 describes these parameters, their chosen values and how we have chosen the values.

#### 5.3.2. Comparison Outcome

Tables 4, 5, 6, and 7 of [19] provide the results of the 20 comparing algorithms on Movie, Music, Celebrity, and History datasets, respectively. Table

Table 2: Parameter Setting of the Proposed PEHER System

Description	Notation	Value	Way of Choosing the Value
Damping factor in (13)	$d$	0.85	Following [27]
Threshold of convergence in (13)	$\tau$	0.001	Based on ad-hoc experiments
The number of iterations in (13)	$r_{max}$	1000	Based on ad-hoc experiments
Balancing factor in (14)	$\gamma$	0.5	Following [38]
Balancing parameter in Algorithm (2)	$w_r$	0.5	Following [19]

3 shows the results of PEHER along with numbers placed within parentheses. These numbers indicate the number of comparing cases (out of 20) for which PEHER has outperformed the comparing one. The last column of Table 3 lists the number of cases for which PEHER has outperformed the comparing one on the corresponding dataset. For example, the entry corresponding to the first row and the last column is 99/100. It means that PEHER has outperformed the comparing methods in 99 out of 100 ( $= 20 \text{ methods} \times 5 \text{ measures}$ ) comparing cases on Movie dataset. The last row of Table 3 lists the number of cases for which PEHER has outperformed the comparing one with respect to the associated measures. For instance, the entry corresponding to the last row and the first column is 80/80. It indicates that PEHER has outperformed the comparing methods in 80 out of 80 comparing scenarios with respect to MRR. An entry of 368/400 corresponding to the last row and the last column denotes that PEHER has outperformed the comparing methods in 368 out of 400 ( $= 20 \text{ methods} \times 4 \text{ datasets} \times 5 \text{ measures}$ ) cases. In other words, the proposed method has performed the best in 92.00% (368 out of 400) cases.

PEHER has performed the best on Music dataset (performed the best

Table 3: Experimental Results of the Proposed PEHER System

Dataset	MRR (#B)	P@30 (#B)	R@30 (#B)	Accuracy (#B)	MSC@30 (#B)	#B/#T
Movie	0.0947 (20)	0.0049 (20)	0.2001 (20)	0.9041 (19)	0.4015 (20)	99/100
Music	0.0874 (20)	0.0060 (20)	0.1837 (20)	0.9017 (20)	0.4016 (20)	100/100
Celebrity	0.1437 (20)	0.0049 (20)	0.2023 (17)	0.8833 (20)	0.4073 (20)	97/100
History	0.2080 (20)	0.0306 (12)	0.2969 (11)	0.9825 (18)	0.6237 (11)	72/100
#B/#T	80/80	72/80	68/80	77/80	71/80	368/400

#B: The number of best performing cases.

#T: The number of total comparing cases in the corresponding scenario.

for 100.00% cases) and the worst on History dataset (performed the best for 72.00% cases). When we use MRR and R@30 for comparison, PEHER performs the best and the worst, respectively. Particularly, when MRR and R@30 are used, PEHER is found to outperform the comparing methods for 100.00% (80 out of 80) and 85.00% (68 out of 80) cases, respectively.

#### 5.4. Experimental Validation of Different Philosophies in PEHER

##### 5.4.1. Validation of Different Philosophies in Preferability Estimator

To inspect how different philosophies in the preferability estimator affect the proficiency, we compare four schemes: (i) a baseline method, (ii) an intra-profile (IntraP) based method, (iii) an inter-profile (InterP) based method, and (iv) the proposed proficiency method, as discussed next.

In the baseline method, we use the question similarity score discussed in Section (4.3.1) for expertise estimation. Considering  $q\_pref[\cdot] = 1$  in (15), we obtain the expertise score of an answerer  $a$  for a new question  $\hat{q}$  as

$$\mathcal{E}(a, \hat{q})_{\text{baseline}} = \sum_{q \in Q_a} \text{sim}[\hat{q}, q]. \quad (16)$$

In the second method, IntraP, we consider the intra-profile preference to determine the question preferability score. Here, we use IntraP proficiency

score as the expertise score of an answerer  $a$  for a new question  $\hat{q}$  as

$$\mathcal{E}(a, \hat{q})_{\text{IntraP}} = \sum_{q \in Q_a} \text{sim}[\hat{q}, q] \times \text{q\_pref}[a, q]_{\text{IntraP}}; \quad (17)$$

$$\text{q\_pref}[a, q]_{\text{IntraP}} = (1/|q|) \sum_{t \in q, q \in Q_a} \text{pref}[a, t]_{\text{IntraP}} \times \text{imp}[t], \quad (18)$$

where,  $\text{sim}[\hat{q}, q]$ ,  $\text{pref}[a, t]_{\text{IntraP}}$ , and  $\text{imp}[t]$  are in (14), (2), and (1), respectively.

In the third method, InterP, we consider inter-profile preference to determine the question preferability. Here, we use InterP proficiency score as the expertise score of an answerer  $a$  for a new question  $\hat{q}$  as

$$\mathcal{E}(a, \hat{q})_{\text{InterP}} = \sum_{q \in Q_a} \text{sim}[\hat{q}, q] \times \text{q\_pref}[a, q]_{\text{InterP}}; \quad (19)$$

$$\text{q\_pref}[a, q]_{\text{InterP}} = (1/|q|) \sum_{t \in q, q \in Q_a} \text{pref}[a, t]_{\text{InterP}} \times \text{imp}[t], \quad (20)$$

where,  $\text{sim}[\hat{q}, q]$ ,  $\text{pref}[a, t]_{\text{InterP}}$  and  $\text{imp}[t]$  are in (14), (4), and (1), respectively.

In the fourth method, proficiency, we use (15) as the expertise score. Table 4 shows the performance improvements of the last three methods (IntraP, InterP, and proficiency) compared to the first (baseline) scheme. Each entry in Table 4 also includes the percentage improvement compared to the baseline. To test the improvement of each method compared to the baseline, we perform  $t$ -test (two-tailed) at the statistical level of significance  $\alpha = 0.001$ . The symbol of dagger ( $\dagger$ ) corresponding to an entry in Table 4 denotes that the  $p$ -value of the corresponding  $t$ -test is smaller than the chosen  $\alpha$ .

In Table 4, there are total 60 ( $= 4$  datasets  $\times$  5 measures  $\times$  3 methods) cases. From Table 4, we make the following observations. First, in 100.00% (each of the 60) cases, the improvements are positive. Second, in 95.00% (57

Table 4: Results of Different Philosophies in Preferability Estimator

Dataset	Methods	MRR (%I)	P@30 (%I)	R@30 (%I)	Accuracy (%I)	MSC@30 (%I)
Movie	IntraP	0.0706 (+27.16%) <sup>†</sup>	0.0039 (+21.47%) <sup>†</sup>	0.1585 (+21.23%) <sup>†</sup>	0.8767 (+00.35%) <sup>†</sup>	0.3230 (+17.27%) <sup>†</sup>
	InterP	0.0806 (+45.28%) <sup>†</sup>	0.0040 (+26.96%) <sup>†</sup>	0.1656 (+26.70%) <sup>†</sup>	0.8772 (+00.41%) <sup>†</sup>	0.3331 (+20.91%) <sup>†</sup>
	Proficiency	<b>0.0890</b> (+60.45%) <sup>†</sup>	<b>0.0045</b> (+41.88%) <sup>†</sup>	<b>0.1800</b> (+37.67%) <sup>†</sup>	<b>0.8785</b> (+00.55%) <sup>†</sup>	<b>0.3715</b> (+34.85%) <sup>†</sup>
Music	IntraP	0.0740 (+24.11%) <sup>†</sup>	0.0059 (+17.52%) <sup>†</sup>	0.1789 (+16.02%) <sup>†</sup>	0.8829 (+00.23%) <sup>†</sup>	0.3816 (+13.28%) <sup>†</sup>
	InterP	0.0793 (+33.07%) <sup>†</sup>	0.0060 (+20.53%) <sup>†</sup>	0.1833 (+18.88%) <sup>†</sup>	0.8828 (+00.22%)	0.3898 (+15.72%) <sup>†</sup>
	Proficiency	<b>0.0867</b> (+45.51%) <sup>†</sup>	<b>0.0066</b> (+31.93%) <sup>†</sup>	<b>0.1974</b> (+28.00%) <sup>†</sup>	<b>0.8839</b> (+00.34%) <sup>†</sup>	<b>0.4199</b> (+24.66%) <sup>†</sup>
Celebrity	IntraP	0.0949 (+33.64%) <sup>†</sup>	0.0042 (+22.11%) <sup>†</sup>	0.1820 (+23.41%) <sup>†</sup>	0.8680 (+00.19%) <sup>†</sup>	0.3588 (+19.35%) <sup>†</sup>
	InterP	0.1029 (+44.79%) <sup>†</sup>	0.0043 (+23.86%) <sup>†</sup>	0.1837 (+24.57%) <sup>†</sup>	0.8686 (+00.26%)	0.3624 (+20.56%) <sup>†</sup>
	Proficiency	<b>0.1144</b> (+61.07%) <sup>†</sup>	<b>0.0048</b> (+37.54%) <sup>†</sup>	<b>0.2004</b> (+35.88%) <sup>†</sup>	<b>0.8700</b> (+00.41%)	<b>0.3903</b> (+29.84%) <sup>†</sup>
History	IntraP	0.1083 (+34.37%) <sup>†</sup>	0.0257 (+23.92%) <sup>†</sup>	0.2467 (+22.64%) <sup>†</sup>	0.9513 (+00.17%) <sup>†</sup>	0.5395 (+15.78%) <sup>†</sup>
	InterP	0.1222 (+51.57%) <sup>†</sup>	0.0274 (+32.01%) <sup>†</sup>	0.2618 (+30.13%) <sup>†</sup>	0.9517 (+00.21%) <sup>†</sup>	0.5633 (+20.88%) <sup>†</sup>
	Proficiency	<b>0.1356</b> (+68.11%) <sup>†</sup>	<b>0.0294</b> (+41.61%) <sup>†</sup>	<b>0.2816</b> (+39.98%) <sup>†</sup>	<b>0.9523</b> (+00.27%) <sup>†</sup>	<b>0.5951</b> (+27.72%) <sup>†</sup>

%I: Percentage improvement; <sup>†</sup>p-value < .001.

out of 60) cases, the improvements are statistically significant. Third, for every dataset and for every measure, the performance improvement for the proficiency method is more than that of both IntraP and InterP methods. As the proficiency method includes both the philosophies of IntraP and InterP schemes, from this limited experiments, we validate that the proposed way to assess proficiency incorporates both the intra-profile preference and the inter-profile preference in an effective manner.

#### 5.4.2. Validation of Different Philosophies in Authority Estimator

We compare the proposed authority estimator in (13) with CBEN [12] using PageRank [27] algorithm as the baseline. Here, we use (13) as the expertise score. We provide the results of the proposed authority estimator on four datasets in Table 5. In the same table, we also present the percentage of improvements within parenthesis after each entry. Form Table 5, we observe that the proposed authority estimator has outperformed the corresponding baseline method in 95.00% (19 out of 20) cases. We also perform *t*-test (two-

Table 5: Results of the Proposed Authority Esimator

Dataset	MRR (%I)	P@30 (%I)	R@30 (%I)	Accuracy (%I)	MSC@30 (%I)
Movie	<b>0.0683</b> (+15.71%) <sup>†</sup>	<b>0.0045</b> (+02.10%)	<b>0.1890</b> (+05.95%) <sup>†</sup>	<b>0.8739</b> (+00.33%) <sup>†</sup>	<b>0.3815</b> (+03.86%) <sup>†</sup>
Music	<b>0.0377</b> (+20.70%) <sup>†</sup>	<b>0.0037</b> (+06.24%)	<b>0.1132</b> (+27.03%) <sup>†</sup>	<b>0.8551</b> (+00.31%) <sup>†</sup>	<b>0.2784</b> (+13.81%) <sup>†</sup>
Celebrity	<b>0.1134</b> (+18.66%) <sup>†</sup>	<b>0.0040</b> (+04.40%)	<b>0.1832</b> (+09.38%) <sup>†</sup>	<b>0.8546</b> (+00.50%) <sup>†</sup>	<b>0.3661</b> (+05.23%)
History	<b>0.1829</b> (+00.87%) <sup>†</sup>	<b>0.0285</b> (+00.88%)	<b>0.2842</b> (+03.73%) <sup>†</sup>	0.9738 (-00.49%) <sup>†</sup>	<b>0.6054</b> (+01.57%) <sup>†</sup>

%I: Percentage improvement; [†]  $p$ -value < .001.

tailed), in the same way as performed in Section 5.4.1. It shows that the improvements of the proposed authority estimator are statistically significant in 78.95% (15 out of 19) cases. However, the only case of deterioration has also been found to be significant. This experiment empirically validates the assumptions behind the proposed authority estimator.

#### 5.4.3. Validation of Different Philosophies in Expertise Estimator

To investigate the effectiveness of the proposed fusion strategy, we compare the proposed PEHER system with the following four methods:

1. Proficiency: Here, we use (15) as the expertise score. Note that, this method has been examined in Section (5.4.1).
2. Authority: Here, we use (13) as the expertise score. It represents the proposed authority estimator examined in Section (5.4.2).
3. PEHER-max( $\cdot$ ): PEHER with the  $\max(\cdot)$  function on the ranks as the fusion technique, instead of using Algorithm 2.
4. PEHER-RRF: PEHER with the RRF [31] on the ranks as the fusion technique, instead of using Algorithm 2.

Table 6 shows the improvements of the proposed PEHER over with these four cases. In Table 6, we have 80 (= 4 datasets  $\times$  5 measures  $\times$  4 strategies)

Table 6: Improvements (%) of the Proposed Fusion Strategy.

Dataset	Comparing Strategy	MRR	P@30	R@30	Accuracy	MSC@30
Movie	Proficiency	+06.34% <sup>†</sup>	+09.23% <sup>†</sup>	+11.16%	+02.92% <sup>†</sup>	+08.09%
	Authority	+36.61%	+10.65% <sup>†</sup>	+05.86% <sup>†</sup>	+03.45% <sup>†</sup>	+05.25%
	PEHER-max(.)	+03.04% <sup>†</sup>	+03.86% <sup>†</sup>	+03.50%	+00.76%	+03.22%
	PEHER-RRF	+07.09%	00.00%	00.00%	00.00%	00.00%
Music	Proficiency	+00.73%	-09.06% <sup>†</sup>	-06.93% <sup>†</sup>	+02.01% <sup>†</sup>	-04.35%
	Authority	+131.8% <sup>†</sup>	+64.37% <sup>†</sup>	+62.23% <sup>†</sup>	+05.44% <sup>†</sup>	+44.26% <sup>†</sup>
	PEHER-max(.)	+01.33% <sup>†</sup>	+02.26%	+02.70%	+00.72% <sup>†</sup>	+01.62%
	PEHER-RRF	+16.05% <sup>†</sup>	-00.15%	-00.33%	00.00%	-00.23%
Celebrity	Proficiency	+25.57% <sup>†</sup>	+03.06%	+00.96%	+01.54%	+04.35%
	Authority	+26.69% <sup>†</sup>	+21.69% <sup>†</sup>	+10.42%	+03.37% <sup>†</sup>	+11.26%
	PEHER-max(.)	+02.10% <sup>†</sup>	+05.76% <sup>†</sup>	+03.45%	+00.41%	+05.33% <sup>†</sup>
	PEHER-RRF	+09.97% <sup>†</sup>	00.00%	00.00%	00.00%	00.00%
History	Proficiency	+53.42% <sup>†</sup>	+03.93% <sup>†</sup>	+05.44% <sup>†</sup>	+03.17% <sup>†</sup>	+04.80% <sup>†</sup>
	Authority	+13.70% <sup>†</sup>	+07.36% <sup>†</sup>	+04.45% <sup>†</sup>	+00.89% <sup>†</sup>	+03.03% <sup>†</sup>
	PEHER-max(.)	+00.73% <sup>†</sup>	+01.40% <sup>†</sup>	+01.43%	+00.14% <sup>†</sup>	+00.95%
	PEHER-RRF	+10.89% <sup>†</sup>	00.00%	+00.01%	00.00%	00.00%

<sup>†</sup> p-value < .001.

cases. The case with the proposed fusion strategy is found to perform better in 77.50% (62 out of 80) cases. Specifically, the proposed system is found to perform either better or equal in 92.50% (74 out of 80) cases. Similar to our previous experiments, we perform pairwise two-tailed *t*-test here and put the symbol of dagger (<sup>†</sup>) in Table 6, when a significant difference is observed. We find that in 41 cases the improvements are statically significant. On the other hand, in only 2 cases the deteriorations are significant.

## 6. Conclusion

Here, we introduce a preference enhanced hybrid expertise retrieval system (PEHER) in CQA services. It incorporates the preference of an answerer regarding her practice of terms. PEHER uses both the intra- and inter-profile preferences along with the importance of terms to determine the preferability of an answerer for archived questions. Again, PEHER utilizes a CBEN

[12], and incorporates textual information to assess the authority of a given answerer. Here, we assume that an answerer is familiar to a given question if she has answered a large number of similar questions. The modified CBEN realizes this familiarity as the weights of the links. To estimate the expertise of an answerer, we assess the proficiency of the answerer, and then, combine her proficiency with her authority.

We compare the proposed method with 20 existing methods on 4 datasets using 5 performance measures. It shows that PEHER performs the best in 92.00% (368 out of 400) cases. We also perform detailed experiments to validate the usefulness of the philosophies incorporated in PEHER. These experiments empirically validate the effectiveness of the philosophies.

Now, we conclude the applicability of the proposed system. First, it considers the preferences of a user when recommending her a new question, which helps to keep an answerer's interest alive in sharing knowledge. Second, the proposed system is expected to reduce the number of unanswered questions, which would enhance an asker's interest to post more questions. Thus, the proposed method may enhance the attachments of both answerers and askers, which is likely to enhance the growth of the associated CQA service. Nevertheless, this work has several possible application areas other than CQA services. For instance, consider a reviewer recommendation system for a journal. Here, the questions, the asker, and the answerers are analogous to the manuscripts, the authors, and the reviewers, respectively. Here, an editor may use the system to get a list of possible good reviewers from a large set of reviewers. It would be more helpful in the cases of mega-journals. This is so because mega-journals usually publish a large number of papers from

diverse domains. Hence, they need to maintain a large number of trustworthy reviewers with diverse expertise. Similarly, this work may be helpful in researcher recommender systems [44], academic paper recommender systems [45], and scholar-friend recommender systems [46].

PEHER has some shortcomings. First, The preference of an answerer as defined in this work is purely based on her answering history. Though history is a proxy for preferences, it may not be a complete one. When a small number of questions appear on the preferred topics of an answerer, PEHER may assign her a low expertise score. This issue may be addressed if we use synonyms of terms (words) while measuring the similarities and preferences. Second, for each new question, PEHER needs to estimate its similarity with all the archived questions. This is computationally expensive, particularly with a large archive. Third, we do not check the sensitivity of the five parameters used in this work. However, as we use the same set of parameter values throughout the experiments and achieve satisfactory performance, the chosen values may be considered as suitable. In future, we plan to investigate how changing these parameter values affect PEHER. We also intend to examine different fusion techniques.

## References

- [1] X. Liu, W. B. Croft, M. Koll, Finding experts in community-based question-answering services, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM, 2005, pp. 315–316.
- [2] J. Guo, S. Xu, S. Bao, Y. Yu, Tapping on the potential of q&a com-

- munity by recommending answer providers, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, ACM, 2008, pp. 921–930.
- [3] B. Li, I. King, Routing questions to appropriate answerers in community question answering services, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM, 2010, pp. 1585–1588.
  - [4] B. Li, I. King, M. R. Lyu, Question routing in community question answering: putting category in its place, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, ACM, 2011, pp. 2041–2044.
  - [5] D. P. Mandal, D. Kundu, S. Maiti, Finding experts in community question answering services: A theme based query likelihood language approach, in: 2015 International Conference on Advances in Computer Engineering and Applications, 2015, pp. 423–427.
  - [6] J. Zhang, M. S. Ackerman, L. Adamic, Expertise networks in online communities: Structure and algorithms, in: Proceedings of the 16th International Conference on World Wide Web, WWW '07, ACM, 2007, pp. 221–230.
  - [7] P. Jurczyk, E. Agichtein, Discovering authorities in question answer communities by using link analysis, in: Proceedings of the sixteenth ACM conference on Conference on Information and Knowledge Management, ACM, 2007, pp. 919–922.

- [8] P. Jurczyk, E. Agichtein, 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, ACM, 2007, pp. 845–846.
- [9] H. Zhu, H. Cao, H. Xiong, E. Chen, J. Tian, Towards expert finding by leveraging relevant categories in authority ranking, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11, ACM, 2011, pp. 2221–2224.
- [10] H. Zhu, E. Chen, H. Xiong, H. Cao, J. Tian, Ranking user authority with relevant knowledge categories for expert finding, World Wide Web 17 (5) (2014) 1081–1107.
- [11] D. Schall, F. Skopik, An analysis of the structure and dynamics of large-scale q/a communities, in: J. Eder, M. Bielikova, A. M. Tjoa (Eds.), Advances in Databases and Information Systems, Springer, 2011, pp. 285–301.
- [12] Ç. Aslay, N. O'Hare, L. M. Aiello, A. Jaimes, Competition-based networks for expert finding, in: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2013, pp. 1033–1036.
- [13] M. Shahriari, S. Parekodi, R. Klamma, Community-aware ranking algorithms for expert identification in question-answer forums, in: Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business, i-KNOW '15, ACM, 2015, pp. 8:1–8:8.

- [14] Y. Lu, X. Quan, X. Ni, W. Liu, Y. Xu, Latent link analysis for expert finding in user-interactive question answering services, in: Semantics, Knowledge and Grid, 2009. SKG 2009. Fifth International Conference on, IEEE, 2009, pp. 54–59.
- [15] W.-C. Kao, D.-R. Liu, S.-W. Wang, Expert finding in question-answering websites: a novel hybrid approach, in: Proceedings of the 2010 ACM symposium on applied computing, ACM, 2010, pp. 867–871.
- [16] D.-R. Liu, Y.-H. Chen, W.-C. Kao, H.-W. Wang, Integrating expert profile, reputation and link analysis for expert finding in question-answering websites, *Information processing & management* 49 (1) (2013) 312–329.
- [17] G. A. Wang, J. Jiao, A. S. Abrahams, W. Fan, Z. Zhang, ExpertRank: A topic-aware expert finding algorithm for online knowledge communities, *Decision Support Systems* 54 (3) (2013) 1442–1451.
- [18] L. T. Le, C. Shah, Retrieving people: Identifying potential answerers in community question-answering, *Journal of the Association for Information Science and Technology* 69 (10) (2018) 1246–1258.
- [19] D. Kundu, D. P. Mandal, Formulation of a hybrid expertise retrieval system in community question answering services, *Applied Intelligence* 49 (2) (2019) 463–477.
- [20] Z. Yan, J. Zhou, Optimal answerer ranking for new questions in community question answering, *Information Processing & Management* 51 (1) (2015) 163–178.

- [21] M. Neshati, Z. Fallahnejad, H. Beigy, On dynamicity of expert finding in community question answering, *Information Processing & Management* 53 (5) (2017) 1026–1042.
- [22] X. Cheng, S. Zhu, S. Su, G. Chen, A multi-objective optimization approach for question routing in community question answering services, *IEEE Transactions on Knowledge and Data Engineering* 29 (9) (2017) 1779–1792.
- [23] X. Liu, W. B. Croft, Cluster-based retrieval using language models, in: *Proceedings of the 27th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2004, pp. 186–193.
- [24] D. R. H. Miller, T. Leek, R. M. Schwartz, A hidden Markov model information retrieval system, in: *Proceedings of the 22nd annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 1999, pp. 214–221.
- [25] V. Lavrenko, W. B. Croft, Relevance based language models, in: *Proceedings of the 24th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2001, pp. 120–127.
- [26] J. M. Ponte, W. B. Croft, A language modeling approach to information retrieval, in: *Proceedings of the 21st annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 1998, pp. 275–281.

- [27] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the web., Tech. rep., Stanford InfoLab (1999).
- [28] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, *Journal of the ACM (JACM)* 46 (5) (1999) 604–632.
- [29] Y. Cao, J. J. Liu, S. Bao, H. Li, Research on expert search at enterprise track of trec 2005, *TREC* 2005, 2005.
- [30] J. Zhang, J. Tang, J. Li, Expert finding in a social network, in: International Conference on Database Systems for Advanced Applications, Springer, 2007, pp. 1066–1069.
- [31] M. Zhang, R. Song, C. Lin, S. Ma, Z. Jiang, Y. Jin, Y. Liu, L. Zhao, S. Ma, Expansion-based technologies in finding relevant and new information: Thu trec 2002: Novelty track experiments, NIST Special Publication SP 251 (2003) 586–590.
- [32] K. S. Jones, A statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation* 28 (1972) 11–21.
- [33] M. Ben-Akiva, D. McFadden, K. Train, Foundations of stated preference elicitation: Consumer behavior and choice-based conjoint analysis, *Foundations and Trends in Econometrics* 10 (1-2) (2019) 1–144.
- [34] M. Danaf, F. Becker, X. Song, B. Atasoy, M. Ben-Akiva, Online discrete choice models: Applications in personalized recommendations, *Decision Support Systems* 119 (2019) 35 – 45.

- [35] F. Becker, M. Danaf, X. Song, B. Atasoy, M. Ben-Akiva, Bayesian estimator for logit mixtures with inter- and intra-consumer heterogeneity, *Transportation Research Part B: Methodological* 117 (2018) 1 – 17.
- [36] S. Chang, A. Pal, Routing questions for collaborative answering in community question answering, in: *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ACM, 2013, pp. 494–501.
- [37] Z. Zhao, Q. Yang, D. Cai, X. He, Y. Zhuang, Expert finding for community-based question answering via ranking metric network learning, in: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, AAAI Press, 2016, pp. 3000–3006.
- [38] K. Balog, L. Azzopardi, M. De Rijke, Formal models for expert finding in enterprise corpora, in: *Proceedings of the 29th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2006, pp. 43–50.
- [39] R. Yeniterzi, Effective and efficient approaches to retrieving and using expertise in social media, *SIGIR Forum* 49 (2) (2016) 152–153.
- [40] J. Zhang, M. S. Ackerman, L. Adamic, Expertise networks in online communities: structure and algorithms, in: *Proceedings of the 16th International Conference on World Wide Web*, ACM, 2007, pp. 221–230.
- [41] L. Wang, B. Wu, J. Yang, S. Peng, Personalized recommendation for new questions in community question answering, in: *2016 IEEE/ACM*

International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2016, pp. 901–908.

- [42] F. Riahi, Z. Zolaktaf, M. Shafiei, E. Milios, Finding expert users in community question answering, in: Proceedings of the 21st International Conference on World Wide Web, WWW '12, ACM, 2012, pp. 791–798.
- [43] J. Liu, B. Jia, H. Xu, B. Liu, D. Gao, B. Li, A topicrank based document priors model for expert finding, in: M. Fei, S. Ma, X. Li, X. Sun, L. Jia, Z. Su (Eds.), Advanced Computational Methods in Life System Modeling and Simulation, Springer, 2017, pp. 334–341.
- [44] Q. Wang, J. Ma, X. Liao, W. Du, A context-aware researcher recommendation system for university-industry collaboration on r&d projects, Decision Support Systems 103 (2017) 46 – 57.
- [45] J. Son, S. B. Kim, Academic paper recommender system using multilevel simultaneous citation networks, Decision Support Systems 105 (2018) 24 – 33.
- [46] Y. Xu, D. Zhou, J. Ma, Scholar-friend recommendation in online academic communities: An approach based on heterogeneous network, Decision Support Systems 119 (2019) 1 – 13.



**Dipankar Kundu** received the Bachelor of Computer Applications (Honors) degree from the University of Burdwan, India, and the Master of Computer Applications degree from Indira Gandhi National Open University, India, in 2004 and 2009, respectively. He is currently pursuing the Ph.D. degree at the University of Calcutta, India. He was a project linked person in the Machine Intelligence Unit at the Indian Statistical Institute, Kolkata, India, from 2012 to 2018. His current research interests include Expertise Retrieval, Social Networking, Information retrieval, and Recommender Systems.



**Rajat Kumar Pal** received the B.E. degree in Electrical Engineering from the Bengal Engineering College, Shibpur under the University of Calcutta, India, and the M.Tech. degree in Computer Science and Engineering from the University of Calcutta, India in 1985 and 1988, respectively. He received the Ph.D. degree from the Indian Institute of Technology (IIT), Kharagpur, India in 1996. Presently, he is working as a Professor with the Department of Computer Science and Engineering, University of Calcutta, India. He holds several international patents and his research interests include VLSI design, Graph theory, Design and analysis of algorithms, Logic synthesis, Computational geometry, etc. Dr. Pal has published more than 200 technical research articles, and authored and co-authored several books.



**Deba Prasad Mandal** was born in 1963 at a village named Bagula in West Bengal, India. He obtained the B.Sc.(Honors) degree in Statistics from Kalyani University, West Bengal, India in 1984. In 1988, he received the Master of Computer Applications (MCA) degree from Jawaharlal Nehru University, New Delhi. He obtained the Ph.D. degree (in Computer Sciences) from Indian Statistical Institute, Kolkata in 1993. From February 1994 to March 1996, he visited the Department of Industrial Engineering, University of Osaka Prefecture, Japan with a Japanese Government Postdoctoral Fellowship. Dr. Mandal received the Young Scientist Award in Computer Sciences from Indian Science Congress Association in 1992. At present, he is an Associate Professor in the Machine Intelligence Unit at the Indian Statistical Institute, Kolkata. His research interests mainly include Pattern Recognition, Fuzzy Sets and Systems, Web Mining and Information Retrieval. He has 40 research articles in his credit.

## Highlights

- We propose a preference enhanced hybrid expertise retrieval (PEHER) system in community question answering services.
- PEHER utilizes the textual information to determine both intra-profile and inter-profile preferences of answerers for each term.
- PEHER considers the textual familiarity between each archived question and the profile of each answerer.

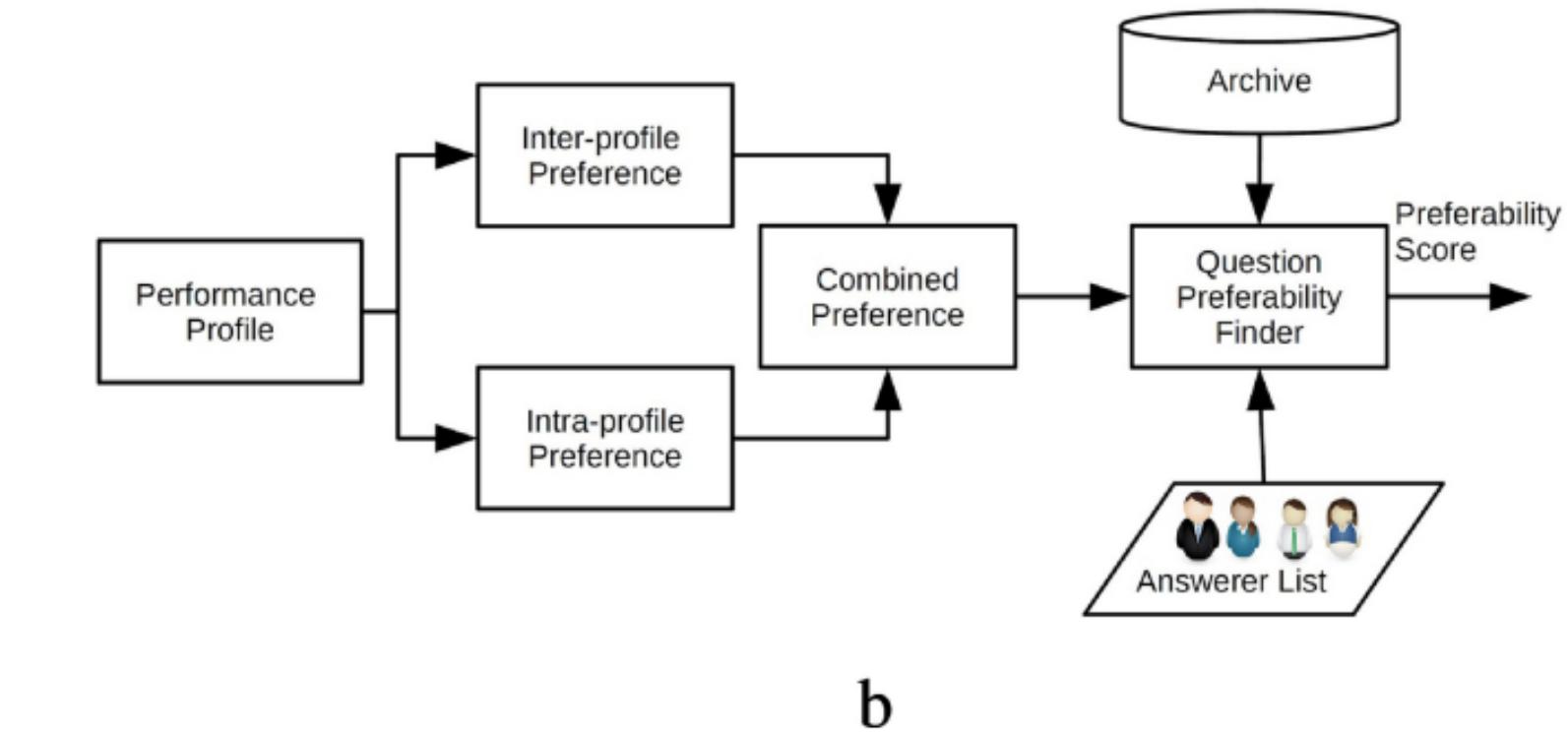
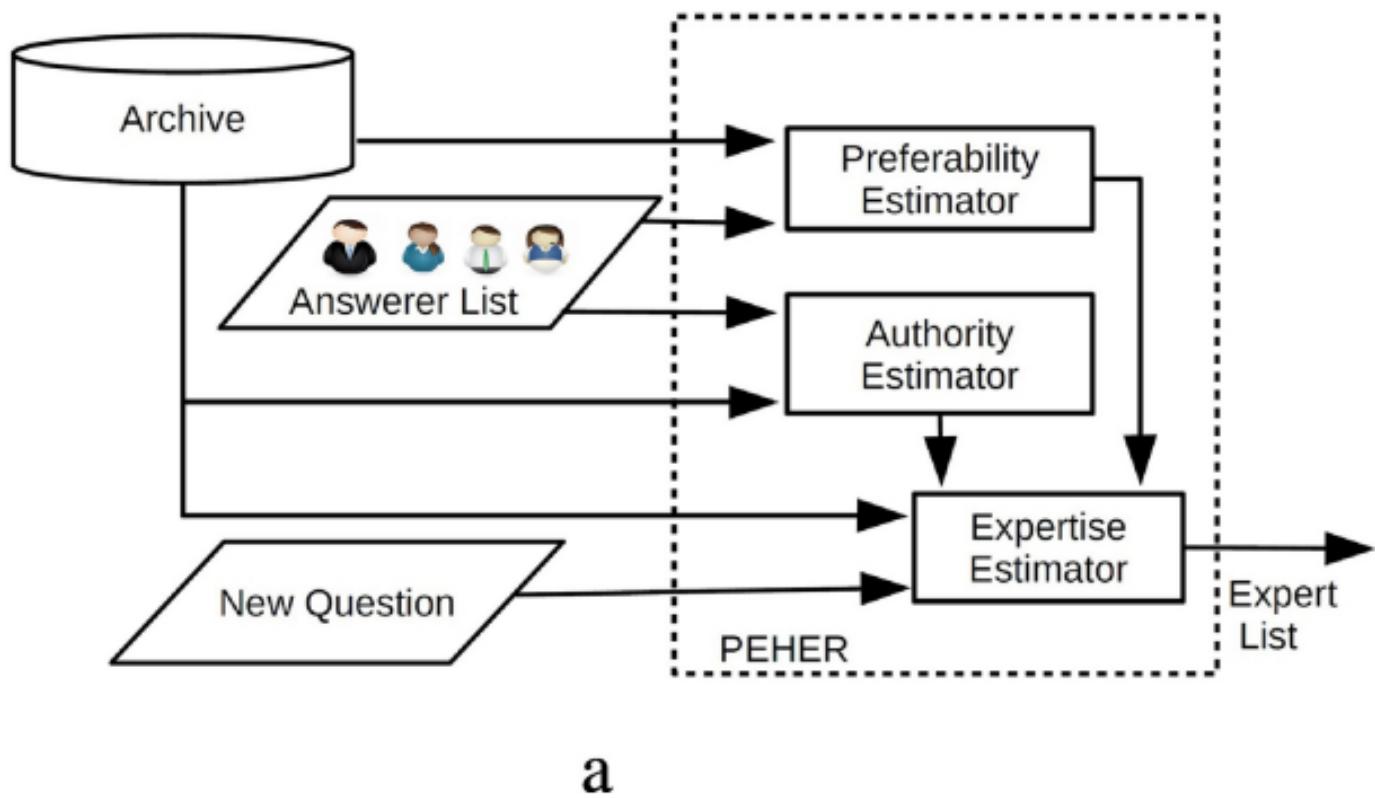


Figure 1

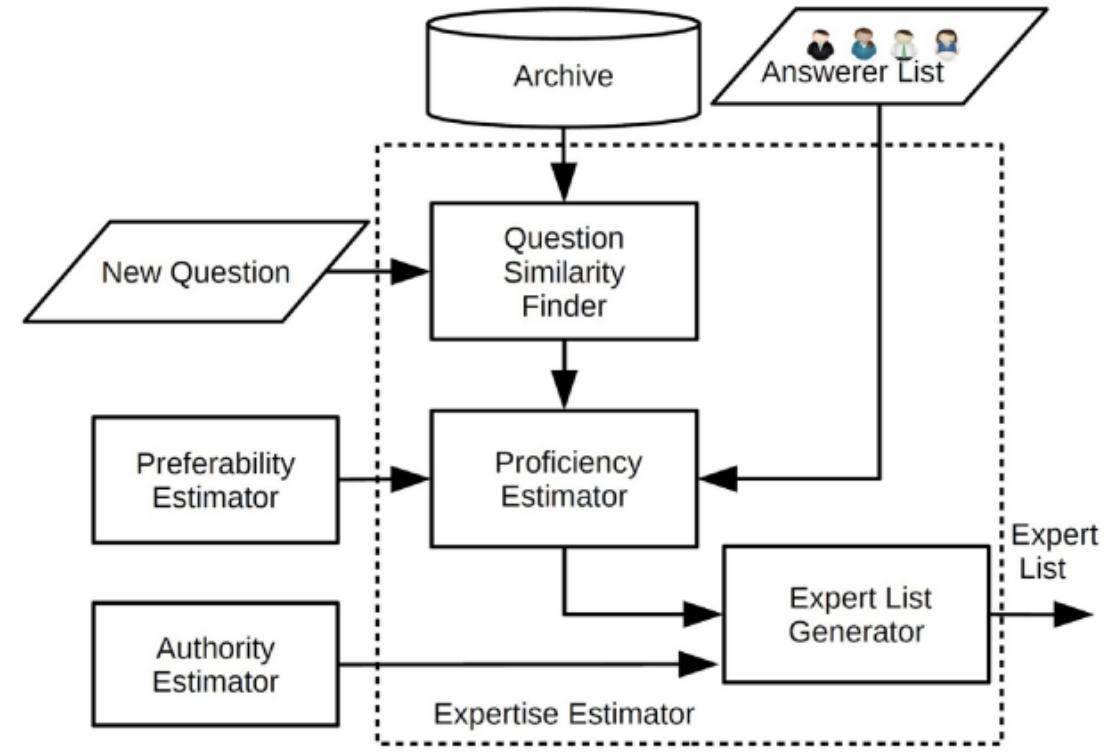
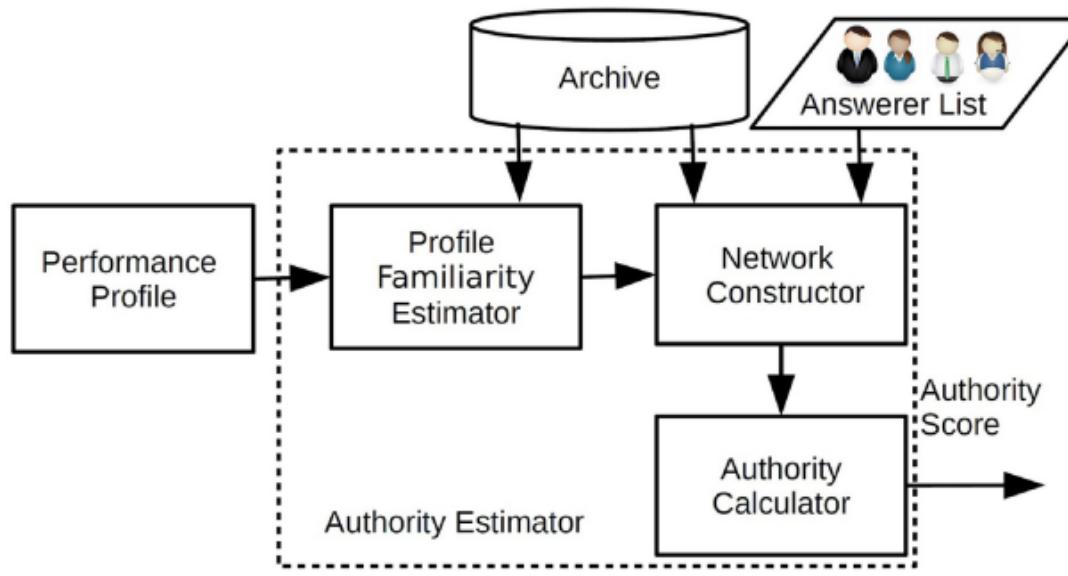


Figure 2