It Takes a Team to Triumph: Collaborative Expert Finding in Community QA Networks

Anonymous Author(s)

ABSTRACT

The increasing complexity and multidisciplinary nature of queries on Community Question Answering (CQA) platforms have rendered the traditional model of individual expert response inadequate. This paper tackles the challenge of identifying *a group of experts* whose combined expertise can effectively address such complex inquiries collaboratively, leading to more accepted answers. Our approach jointly learns topological and textual information extracted from the CQA environment in an end-to-end fashion. Extensive experiments on several real-life datasets indicate that our approach improves the quality of expert ranks on average 4.6% and 7.1% in terms of NDCG and MAP, respectively, compared to the best baseline. The results also reveal that groups formed by our approach are more collaborative and on average 61.6% of members recommended by our approach are among the true answerers of questions which is around 6.1 times improvement compared to the baselines.

CCS CONCEPTS

• **Information systems → Learning to rank**; **Expert search**.

KEYWORDS

Expert Finding, Team Formation, Learn to Ranks, Graph Convolutional Networks, Deep Learning.

ACM Reference Format:

Anonymous Author(s). 2018. It Takes a Team to Triumph: Collaborative Expert Finding in Community QA Networks. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Community-based question answering (CQA) platforms such as *Stack Exchange* and *Quora* play a pivotal role in matching information seekers with knowledgeable individuals. The task of identifying suitable community experts for new questions is crucial as it can significantly enhance the quality and longevity of answers [5]. This process, often referred to as *expert finding* or *question routing* in the literature [59, 66, 68], traditionally relies on the similarity between the textual content of new questions and the historical contributions of experts [8, 22, 54]. Recent approaches also consider the social networks of experts to improve the relevance and quality of matches [35, 50, 59]. Although such approaches achieve promising results, a significant number of questions in CQA systems still remain unanswered or without accepted answers. For example, as of January 2024, there were approximately 3.4 million unanswered questions and over 11.7 million questions without an accepted answer on Stack Overflow[1].

One of the main reason is the complexity of questions which often necessitates input from multiple experts. This is reflected in the average length of discussion threads—6.7 on platforms like StackOverflow and Yahoo![40], indicating that effective answers often result from the collaboration among experts with diverse, complementary point of views. Despite these implications, existing methods frequently overlook the collaborative dynamics among experts, which is a foundational aspect of problem-solving on CQA platforms. Indeed, effective and lasting answers typically emerge from early collaborative interactions among experts, who combine their diverse and complementary skills to provide comprehensive solutions[5]. This highlights a pressing need for methodologies that not only identify individual expertise but also facilitate the formation of expert groups capable of collaboratively addressing more complex and multidisciplinary questions.

To this end, we argue that to effectively capture such interactions and leverage both textual and topological information for expert finding, the learning process must happen in tandem such that the impact of latent topological information is considered during the process of learning representations for textual content and vice versa. To do so, we present a novel *kernel-based* model that jointly learns embedding representations for information within the CQA network based on both topological and textual information. We also incorporate a mechanism to consider the collaboration level of the top-ranked experts for an input question while retrieving experts for a new question. Our proposed method ranks compact subgraphs of the CQA network using their corresponding textual contents given a subgraph built using newly posted questions. Thus, collaborative experts with similar expertise obtain similar ranking scores. In summary, the main contributions of our work are outlined as follows:

- we formulate the problem of *collaboration-driven experts finding in a CQA platform* as a ranking problem that is driven by embedding representations learned by jointly considering both textual and topological information in tandem;
- we systematically consider the level of past collaboration history between the ranked list of retrieved experts such that they show effective past collaboration with each other as opposed to only ranking experts to question based on their relevance independently;

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

[1]https://data.stackexchange.com/stackoverflow/queries

- we propose a novel end-to-end model to learn the latent representations of entities in a CQA environment by satisfying two objectives: (1) employing topological and textual information extracted from the environment, and (2) adopting experts' past collaborations. Our proposed kernel pooling-based model captures the topological features of entities, the textual content in the CQA network, and the experts' collaboration history;
- we carry out extensive comparative experiments to shed light on the performance of our model and the existing state-of-the-art expert finding techniques in terms of finding an effective list of experts.
- we conduct extensive experiments on four real-life datasets to investigate the impact of using different types of information for the expert finding task. Our ablation study reveals that jointly learning latent representations of entities such as questions, owners, tags, and experts in the CQA network along with the content similarity of questions and their answers in an end-to-end framework boosts the quality of the expert finding task in terms of widely-used ranking metrics.

2 RELATED WORK

The literature can be categorized into two main streams: *expert finding*, which focuses on identifying individuals with specific expertise, and *collaborative team discovery*, which aims to assemble groups of experts that can effectively work together to complete a given project.

2.1 Expert Finding

Recent studies on expert finding focus on matching new questions with experts who have the requisite knowledge and willingness to respond [4, 12, 35, 69]. These studies are categorized into three main approaches based on the information sources used in Community Question Answering (CQA) platforms: *content-based*, *network-based*, and *hybrid* methods.

Content-based methods rely on the textual content produced by experts, such as questions, answers, and tags, to recommend the most suitable answerers for new questions. These methods typically frame expert finding as a document ranking problem, using language models [2, 3, 22, 32, 33, 39, 49, 70, 70, 71], learning to rank techniques [7, 22], latent topic modeling [17, 34, 42, 53, 63], and collaborative filtering [9, 28, 58, 64, 69]. Network-based methods leverage the topological information derived from relationships among entities within the environment to identify experts [21, 56, 72]. For example, some studies construct user-user graphs and apply link analysis techniques to measure user authority using centrality metrics like PageRank and HITS [23, 67]. Hybrid methods integrate both content-based and network-based approaches to enhance expert finding by combining textual and topological information from the CQA environment. These methods utilize techniques such as linear combination [36], learning to rank [1, 57], reciprocal rank fusion (RRF) [29, 30], and probabilistic generative models [65].

Traditional methods often depend on hand-crafted features and struggle to capture the semantic depth of questions for expert recommendation. However, recent advancements in deep learning have significantly improved expert finding systems by enabling models to learn semantic features from textual data and extract topological information from CQA environments. For instance, Peng et al. [46] introduced a multi-view matching method that learns features from question titles, bodies, and tags and integrates this information using a personalized attention network. Liu et al. [37] developed a non-sampling learning model that leverages complete data rather than negative sampling. Peng et al. [48] presented the Hierarchical Matching network (EFHM), which includes word and question-level match encoders to capture fine-grained semantic matching and an expert-level match encoder for overall expert feature matching. Sun et al. [59] constructed a heterogeneous network and used a graph convolutional network to learn embeddings of various entities end-to-end. Li et al. [35] applied a heterogeneous information network (HIN) embedding model to embed question content, askers, answerers, and their relationships into a shared latent space, using these representations to rank potential answerers. Qian et al. [50] proposed a model that combines contrastive learning with meta-path walks to learn user interest and expertise embeddings in a unified framework. Our method differs by jointly learning latent representations of both textual and topological information, resulting in enriched embeddings, and using an end-to-end framework based on the quality of predicted expert ranks.

Pretrained language models (PLMs) like BERT [13] have further enhanced language modeling by enabling models to learn generalized knowledge representations from large unsupervised datasets. This knowledge transfer has been particularly beneficial for downstream NLP tasks in low-data scenarios. Inspired by this, researchers have explored pre-training's potential for expert identification [38, 44]. For example, Liu et al. [38] introduced ExpertBert, a pretraining language model designed for expert finding on CQA platforms, which focuses on modeling questions, experts, and their matching patterns. Peng et al. [44] proposed an expert-level pre-training paradigm that integrates expert interest and expertise, incorporating historical answered question titles and vote score information for comprehensive expert representations. They further extended this work with personalized information integration and a fine-grained expert pre-training architecture [47], and a CQA-domain contrastive pre-training framework [45] that improves question representations through a title-body contrastive learning task and a personalized tuning network.

Despite many approaches to expert finding in CQA systems, most ignore the collaborative nature of question answering, overlooking past collaborations among experts. The research by [6] is one of the few that addresses collaborative question routing in CQA. Unlike our end-to-end framework that jointly captures textual and topological information, their proposed greedy algorithm (CQR) uses heuristics to capture expert expertise, availability, and compatibility, forming a collaborative group based on these factors.

Most existing methods for expert finding in CQA systems neglect the collaborative aspect of expert recommendations, making it challenging to provide qualified answers for multidisciplinary questions requiring expertise in various fields. Only a few studies [6, 14] aim to route new questions to small teams of collaborating experts. Our approach differentiates itself by estimating collaboration among experts and computing the similarity between required skills for new questions and the skills of experts. Our method learns and uses embeddings of experts, questions, and tags, assigning higher ranking scores to experts with more interactions or similar tags due to the high similarity in their embeddings. This contrasts with other methods that rely on tags and expert co-occurrence networks to estimate collaborative willingness among experts.

2.2 Collaborative Team Discovery

A related line of research to collaborative expert finding is the problem of assembling expert teams from networks using graph-based search techniques, which has garnered significant attention in recent years [15, 24, 25, 27, 27, 52, 61]. Given a graph whose nodes are experts and edges represent their past collaborations, these methods aim to find a compact subgraph of the network whose nodes possess the skill sets required by the question while optimizing a predefined objective functions. For example, Lappas et al. [31] proposed a method, called `CC` here, to find a team while maximizing the collaboration level among the members. Khan et al. [25] proposed approximation algorithms, called `CS`, to form compact groups in a way that members are closely connected and each one owns as many required skills as possible. Kargar et al. [24] designed a method, called `CO` here, to find a team while maximizing the collaboration level among team members and their expertise level by considering the problem of group discovery over weighted node-labeled network graphs.

Recently, there has been growing interest in using neural-based approaches for the problem of forming expert teams from expert networks [18, 51]. Sapienza et al.'s work [55] is a pioneering effort in this area, utilizing an autoencoder design to accelerate computation. However, this approach tends to overfit, resulting in suboptimal performance, especially given the sparse nature of collaboration networks. Meanwhile, Nikzad-Khasmakhi et al. [43] leverage a neural architecture to learn expert representations, enabling expert retrieval by measuring similarity scores between required skills and experts. Rad et al. [18] take a different approach with a variational Bayesian neural network that goes beyond simple mappings between skill and expert nodes. Their method identifies teams with a history of collaboration, ensuring comprehensive skill coverage.

Despite the advances in team formation methods, these approaches have limitations that hinder their direct application to the collaborative expert finding task: (1) they require a predefined set of skills, and (2) they are computationally expensive (NP-hard in practice) and rely on heuristic-based approximations that can lead to suboptimal teams due to the local exploration of subgraphs. Our method addresses these limitations by mapping the CQA network graph and its textual data into an efficient embedding space.

3 PROBLEM FORMULATION

Let $Q = \{q_1, q_2, ..., q_n\}$ be a set of n questions, and $E = \{e_1, e_2, ..., e_m\}$ be a set of m experts (or answerers). Let $q_e^{(i)}$ be a set, represented by $\{e_1^{(i)}, e_2^{(i)}, ..., e_{n_i}^{(i)}\}$. This set contains n_i answerers ($e_i \in E$) who have answered the question q_i. Their respective answers are given in the set $a^{(i)}$, denoted by $\{a_1^{(i)}, a_2^{(i)}, ..., a_{n_i}^{(i)}\}$, corresponding to the same question q_i. Each answer receives a voting score, $s_j^{(i)}$, which is calculated by taking the difference between the up-votes and down-votes given by users who have seen this answer. Also, allow

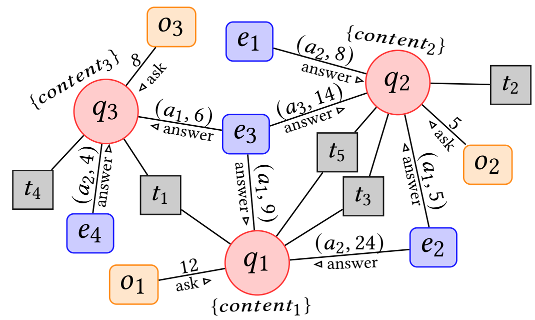

Figure 1: A CQA heterogeneous network

$t^{(i)} = \{t_1^{(i)}, t_2^{(i)}, .., t_{z_i}^{(i)}\}$ to be a set of tags for question q_i assigned by its asker denoted as o_i.

Problem statement. Given a new question q', the problem of finding collaborative experts is to build a ranked list of experts from E that satisfy two main objectives: (**O1**) each recommended expert possesses a high level of expertise related to those needed to answer q_k. Assume that the model $\mathcal{R} : E \rightarrow \mathbb{N}_1$, provides a ranked list of experts for q_k. Given two experts, e_i and e_j, answering a question q_k. If $s_i^{(k)} \geq s_j^{(k)}$, then the ranking score of e_i for q_k will also be greater than or equal to that of e_j:

$$\mathcal{R}(e_i|q_k) \geq \mathcal{R}(e_j|q_k)$$

Here, $\mathcal{R}(e|q_k)$ denotes a positive integer, representing the ranking score of expert e for q_k.

(**O2**) Each expert is highly inclined to collaborate. For two ranked lists, Π_m and Π_n, Π_m is preferred if:

$$\sum_{e_i, e_j \in \Pi_m} \mathsf{P}_{\text{cl}}(e_i, e_j) \geq \sum_{e_k, e_l \in \Pi_n} \mathsf{P}_{\text{cl}}(e_k, e_l)$$

Here, P_{cl} represents past collaborations between two experts; greater past collaboration indicates a higher likelihood of future cooperation.

In this paper, we model our data as a network. The Community Question Answering (CQA) heterogeneous network is denoted as $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V}$ is the set of node, $\mathcal{E}$ represents the set of edges, and $\mathcal{T}$ denotes the types of nodes and edges, respectively. In this graph, nodes can be *questions (q)*, *tag (t)*, *asker (o)*, or *experts (e)*. Edges can also be of types: question-expert (q–e), question-asker (q–o), and question-tag (q–t).

Fig. 1 illustrates a CQA heterogeneous network. The weight on an edge (for $q - o$, $q - w$ edges) shows the voting score. For instance, the weight of 4 between a_2 and e_4 shows that the answer a_2 from expert e_4 received 4 votes. Additionally, question q_3 is associated with two tags: t_4 and t_1.

4 PROPOSED APPROACH

Figure 2 shows an overview of the proposed framework. Given a new question, we leverage both the structure and semantics of the CQA heterogeneous network to identify experts. We first introduce a sub-graph kernel pooling technique, encoding the network's topological data while factoring in historical collaborations between experts (explained in Subsection 4.1). Subsequently, we extract and encode the semantics from the attributes of the nodes and

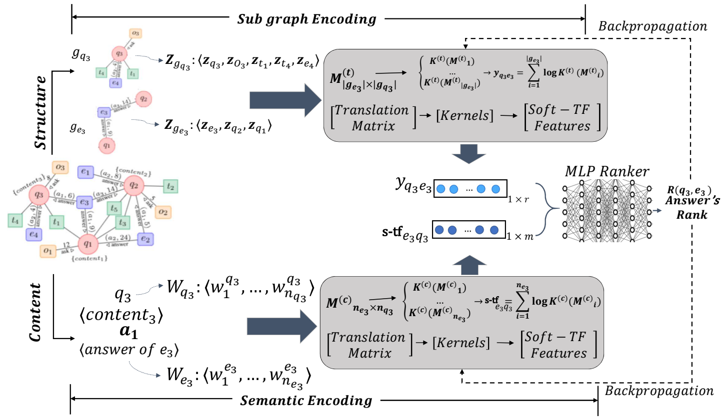

Figure 2: The proposed framework overview.

edges within sub-graphs using another kernel pooling approach (explained in Subsection 4.2). These processed features are then input into a multilayer perceptron, determining the expert rankings (explained in Subsection 4.3). Our model undergoes end-to-end training and, once trained, can rank experts for any new question (explained in Subsection 4.4).

4.1 Sub-graph Encoding

To capture the topological information within the CQA network, our method begins by iterating over each node q categorized as a 'question' in network $\mathcal{G}$. For each question node, we construct sub-graphs: g_q (derived from the question node) and g_e (sourced from either the linked or unlinked answerer node to the question). These sub-graphs are obtained via a random walk initiated from the respective question and answerer nodes in the network. Inspired by document ranking techniques [10, 62], we treat the question subgraph as a query. Concurrently, its linked answerer subgraph is viewed as a relevant document, while some unlinked answerer subgraphs (selected randomly from the network) to the question are considered irrelevant documents.

Given a question and its associated answerer subgraphs, the node embeddings within the sub-graphs g_q and g_e can be formulated as:

$$\mathbf{Z}_{g_q} = (\mathbf{z}_1^q, \mathbf{z}_2^q, ..., \mathbf{z}_{|g_q|}^q)$$

and

$$\mathbf{Z}_{g_e} = (\mathbf{z}_1^e, \mathbf{z}_2^e, ..., \mathbf{z}_{|g_e|}^e)$$

The dimensions of $\mathbf{Z}_{g_q}$ and $\mathbf{Z}_{g_e}$ are defined as $\mathbb{R}^{|g_q|\times d}$ and $\mathbb{R}^{|g_e|\times d}$, respectively, where d is the embedding dimension of nodes. Notably, $|g|$ represents the number of nodes in sub-graph g.

To obtain the similarity feature vector for the pair (g_q, g_e), we employ a differentiable function, Φ, defined as:

$$\mathbf{y}_{qe} = \Phi(\mathbf{Z}_{g_q}, \mathbf{Z}_{g_e}), \tag{1}$$

where $\mathbf{y}_{qe} \in \mathbb{R}^r$ denotes the dimension of the feature vector, capturing the topological similarity between g_q and g_e.

We define function Φ as a neural learn-to-ranking method based on *kernel pooling* as follows. First, embedding vectors of nodes in g_q and g_e are utilized to compute the translation matrix $\mathbf{M}^{(t)} \in \mathbb{R}^{|g_e|\times|g_q|}$ as:

$$m_{ij}^{(t)} = \frac{\mathbf{z}_i^e \mathbf{z}_j^q}{||\mathbf{z}_i^e||.||\mathbf{z}_j^q||}, \tag{2}$$

where embedding vectors $\mathbf{z}_i^e$ and $\mathbf{z}_j^q$ are the embedding vectors of node i in g_e and node j in g_q, respectively. Such embedding vectors of nodes are randomly initialized and learned in an end-to-end manner during the training phase.

Element $m_{ij}^{(t)}$ reveals the similarity between embedding vector of node i in g_e and node j in g_q. Let $\mathbf{M}_i^{(t)}$ denote row i in matrix $\mathbf{M}^{(t)}$ in which it captures the similarity between node i in g_e and all nodes in g_q. We reduce the size of $\mathbf{M}_i^{(t)}$ $(i = 1, 2, ..., |g_e|)$ from $|g_q|$ to r using r kernels to obtain an r-dimensional feature vector. The effectiveness of the feature vector depends on the kernels used in the model. The RBF kernel is adopted as follows due to the fact that it is differentiable and its high performance reported in the literature [62].

Suppose r RBF kernels as $\mathbf{K}^{(t)} = \{K_1^{(t)}, K_2^{(t)}, ..., K_r^{(t)}\}$, kernel $K_k^{(t)}$ $(k = 1, 2, ..., r)$ is applied on the i^{th} row of the translation matrix $\mathbf{M}^{(t)}$, i.e., $\mathbf{M}_i^{(t)}$, as follows:

$$K_k^{(t)}(\mathbf{M}_i^{(t)}) = \sum_{j=1}^{|g_q|} exp\left(-\frac{(m_{ij}^{(t)} - \mu_k)^2}{2\sigma_k^2}\right). \tag{3}$$

where μ_k and σ_k are the parameters of kernel $K_k^{(t)}$, and $K_k^{(t)}(\mathbf{M}_i^{(t)}) \in \mathbb{R}$. By applying r kernels on $\mathbf{M}_i^{(t)}$, we build a vector with r values as:

$$\mathbf{K}^{(t)}(\mathbf{M}_i^{(t)}) = \left(K_1^{(t)}(\mathbf{M}_i^{(t)}), K_2^{(t)}(\mathbf{M}_i^{(t)}), ..., K_r^{(t)}(\mathbf{M}_i^{(t)})\right). \tag{4}$$

Given matrix $\mathbf{M}^{(t)}$ with $|g_e|$ rows, we end up with $|g_e|$ vectors as $\mathbf{K}^{(t)}(\mathbf{M}_1^{(t)}), \mathbf{K}^{(t)}(\mathbf{M}_2^{(t)}), ..., \mathbf{K}^{(t)}(\mathbf{M}_{|g_e|}^{(t)})$. Finally, the log-sum of such $|g_e|$ feature vectors is computed to obtain the similarity between subgraphs g_e and g_q as feature vector $\mathbf{y}_{qe}$ as:

$$\mathbf{y}_{qe} = \sum_{i=1}^{|g_e|} log\ \mathbf{K}^{(t)}(\mathbf{M}_i^{(t)}). \tag{5}$$

where $\mathbf{y}_{qe} \in \mathbb{R}^r$.

4.2 Semantic Encoding

Parallel to the sub-graph encoding, the similarity between the contents of attributes of the question node and its connected answerer is captured by the semantic encoding component as follows. Suppose txt_q and txt_e with n_q and n_e words are the title and body of starting question q in g_q and the content of the answer written by staring expert e in g_e, respectively. Let $\mathbf{W}_q = (\mathbf{w}_1^q, \mathbf{w}_2^q, ..., \mathbf{w}_{n_q}^q)$ and $\mathbf{W}_e = (\mathbf{w}_1^e, \mathbf{w}_2^e, ..., \mathbf{w}_{n_e}^e)$ be lists of word embedding vectors in txt_q and txt_e. Assume that d' is the word embedding dimension in txt_q and txt_e, i.e., $\mathbf{w}_i^q, \mathbf{w}_j^e \in \mathbb{R}^{d'}$. We define a differentiable function Ψ to obtain the similarity of pair (txt_q, txt_e) as:

$$\textbf{s-tf}_{eq} = \Psi(\mathbf{W}_q, \mathbf{W}_e). \tag{6}$$

We utilize a neural learning to rank method based on *kernel pooling* to encode the textual similarity between q and the answer provided by e. First, each element of the translation matrix $\mathbf{M}^{(c)}$ is computed using the embeddings of the words as: $\mathbf{w}_i^e \mathbf{w}_j^q / ||\mathbf{w}_i^e||.||\mathbf{w}_j^q||$. Then, a set of RBF kernels denoted by $\mathbf{K}^{(c)}$ with m kernels as $\{K_1^{(c)}, K_2^{(c)}, ..., K_m^{(c)}\}$ is applied on matrix $\mathbf{M}^{(c)}$ to obtain the soft term frequency feature (s-tf) vector. The RBF kernel $K_k^{(c)}$ ($k = 1, 2, ..., m$) is computed on the i^{th} row of matrix $\mathbf{M}^{(c)}$, i.e., $\mathbf{M}_i{}^{(c)}$, as:

$$K_k^{(c)}(\mathbf{M}_i^{(c)}) = \sum_{j=1}^{n_q} exp\left(-(m_{ij}^{(c)} - \mu_k)^2 / 2\sigma_k^2\right), \tag{7}$$

where μ_k and σ_k are the parameters of kernel $K_k^{(c)}$. Finally, vector $\textbf{s-tf}_{eq} \in \mathbb{R}^m$ is computed as: $\sum_{i=1}^{n_c} log\ \mathbf{K}^{(c)}(\mathbf{M}_i^{(c)})$.

4.3 Expert Rank Predictor

The topological and textual similarities between question q and its answerers encoded in $\mathbf{y}_{qe}$ and $\textbf{s-tf}_{eq}$ are utilized by a multilayer perceptron network to predict the ranking scores of the answerers of the question. Given a question, we define $\mathcal{R}$ as the proposed model to rank experts:

$$\mathcal{R}(q, e) = \sigma^{(n)}\left(\sigma^{(n-1)}(..\sigma^{(1)}(\mathbf{x}\theta^{(1)} + b^{(1)})..)\theta^{(n)} + b^{(n)}\right), \tag{8}$$

where $\mathbf{x} = \mathbf{y}_{qe} || \textbf{s-tf}_{eq}$ and $||$ is the concatenation of two vectors, and $\sigma^{(i)}$ is a non-liner activation function in layer i with trainable parameters $\theta^{(i)}$ and bias $b^{(i)}$. Note that $\mathcal{R}(q, e)$ predicts the voting score of the answer provided by expert e to answer question q.

4.4 Model Training

Consider question $q^{(i)}$ and its true answerers set as $q_{e^+}^{(i)}$. Let $q_{e^-}^{(i)}$ be a set of random negative samples where $|q_{e^-}^{(i)}| = |q_{e^+}^{(i)}|$. The loss value for n questions is defined as:

Algorithm 1: Model Training

Input: $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{T})$.
Output: θ, b, $\mathbf{Z}$, $\mathbf{W}$, Π.
begin
 Randomly initial θ, b, node and word embeddings namely $\mathbf{Z}$, $\mathbf{W}$.
 /* **Training** in each epoch */
 foreach *node q of type question in $\mathcal{G}$* **do**
  q_{e^+}: a set of nodes of type answerers in $\mathcal{G}$ connected to node q_i.
  q_{e^-}: a random answerers in $\mathcal{G}$ disjoined to q with size $|q_{e^+}|$.
  foreach *pair $(e_+, e_-) \in E_q = \{q_{e^+} \times q_{e^-}\}$* **do**
   Build subgraphs g_q, g_{e^+}, and g_{e^-}.
   /*Parallel Computations:*/
   $\mathbf{y}_{qe^+} = \Phi(\mathbf{Z}_{g_q}, \mathbf{Z}_{g_{e^+}})$. using Eq. 1
   $\mathbf{y}_{qe^-} = \Phi(\mathbf{Z}_{g_q}, \mathbf{Z}_{g_{e^-}})$.
   $\textbf{s-tf}_{e^+q} = \Psi(\mathbf{W}_q, \mathbf{W}_{e^+})$ using Eq. 6
   $\textbf{s-tf}_{e^-q} = \Psi(\mathbf{W}_q, \mathbf{W}_{e^-})$
   /*Predict ranks using Eq. 8 */
   $\hat{s}_{e^+} = \mathcal{R}(q, e^+)$.
   $\hat{s}_{e^-} = \mathcal{R}(q, e^-)$.
  end
  Minimize $\mathcal{L} = \sum_{(e^+, e^-) \in E_q} max(0, 1 - \hat{s}_{e^+} + \hat{s}_{e^-}) / |E_q|$.
 end
 /* Collaborative Experts Retrieval */
 Create subgraph $g_{q'}$ for new question q'.
 foreach *expert e in E* **do**
  Compute $\mathbf{y}_{q'e} = \Phi(\mathbf{Z}_{g_{q'}}, \mathbf{Z}_{g_e})$.
  Compute $\textbf{s-tf}_{eq'} = \Psi(\mathbf{W}_{q'}, \mathbf{W}_e)$.
  $\hat{s}_e = \mathcal{R}(\mathbf{y}_{q'e}, \textbf{s-tf}_{eq'})$.
 end
 Π = (top ranked experts based on their $\hat{s}_e$)
end

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{e^+ \in q_{e^+}^{(i)}, e^- \in q_{e^-}^{(i)}} max\ (0, 1 - \hat{s}_{e^+} + \hat{s}_{e^-})}{|q_{e^+}^{(i)}|}, \tag{9}$$

where $\hat{s}_e = \mathcal{R}(q^{(i)}, e)$. The model is trained to minimize the proposed loss function.

Algorithm 1 illustrates the detailed steps to train the model for collaborative expert finding. The algorithm begins by initializing parameters and embeddings, which are then refined through iterative training. Each epoch consists of processing nodes labeled as questions in the graph $\mathcal{G}$, for which paired sets of connected and randomly disconnected answerers are constructed. The algorithm constructs subgraphs for each question and its connected and disconnected answerers, and computes embeddings to predict the relative ranks of these answerers using the defined functions Φ and Ψ. Subsequently, the model is optimized using a margin-based loss function designed to enhance the separation between the scores of connected (positive) and disconnected (negative) answerers, thereby improving the accuracy in identifying true experts. In the final phase, the trained model is applied to retrieve top-ranked experts for new questions. This retrieval process evaluates the relevance of potential experts based on learned representations and ranking functions, ensuring effective identification of suitable collaborators.

Table 1: Statistics of the datasets.

Dataset	N	Questions	Experts+Owners	Tags
android	2,903	882	1,511	510
history	4,102	1,697	1,838	567
dba	6,959	2,906	3,475	578
physics	11,265	4,912	5,636	717

5 EXPERIMENTAL RESULTS

In this section, we present the findings from a series of experiments conducted to evaluate the performance of our proposed model against baselines.

5.1 Datasets and Baselines

Datasets. Table 1 presents a summary of the datasets that we used which were released by Stack Exchange in September 2019. During data preprocessing, stop words and special characters were removed and only questions with a minimum of two answers were retained. Also, answers with voting scores less than four were removed. The pre-processed datasets along with our code and the results are publicly available[2].

Baselines. Two sets of baselines are adopted for investigating the performance of the proposed approach:

- **Expert Finding**: we compare the proposed framework with the state-of-the-art *expert finding* techniques, i.e., `PMEF`[46], `EndCold` (`EnC`) [59], and `NeRank` (`NeR`) [35]. `PMEF`[46] incorporates different view textual information for expert and question learning in a personalized way. The `EnC` method employs a graph convolutional network to learn the topological information of the CQA heterogeneous network in an end-to-end way. The `NeR` method uses both the textual and topological information extracted from a CQA environment to build latent representations. Then, such latent representations are used to rank experts. Furthermore, inspired by [39], we used existing neural-based learning to rank methods including DUET [41], CKNRM [11], KNRM [62], and DSSM[19] to build different expert finding techniques that only rely on the textual information in a CQA environment. We employ the default parameters implemented in MatchZoo [16].
- **Collaborative Team Discovery**: we also compare the results with the collaborative question routing techniques, i.e., `CQR` [5] which models the collaborations among experts using a homophily graph. Then, a greedy technique is employed to build a collaborative group of experts. Furthermore, we compare our work with three team formation techniques, namely `CC` [31], `CO` [24], `CS` [25]. These methods find a set of experts that collectively cover the skills required to answer the new question. These methods employ heuristics to locally explorer on the CQA graph to form a group. Therefore, they are not always able to find the optimal team given the local search on the graph.

[2]https://anonymous.4open.science/r/SIGIRAPColExperts

5.2 Evaluation Metrics

We used two groups of evaluation metrics based on our objectives, namely **O1** and **O2**:

- **Ranking Metrics**: To evaluate methods based on objective **O1**, popular ranking metrics, namely *normalized discounted cumulative gain* (`NDCG`) [20] and *mean average precision* (`MAP`)[60], are employed. The `NDCG@n` indicates how well the discovered ranked list of n experts matches with their true ranks. Similarly, `MAP@n` demonstrates that on average what portion of the top-n ranked experts are among the true experts.
- **Collaboration Metrics**: Several metrics are adapted from the literature to evaluate the quality of discovered collaborative experts for a new question [5] (objective **O2**). Intuitively, answering common past questions is considered a sign of collaboration among recommended experts. We use the following metrics by considering n test questions labeled as $q_1, .., q_n$, and when $\hat{q}_e^{(i)}$, and $q_e^{(i)}$ are the recommended and ground truth answerers of the test question q_i: *(1) Gold Standard Match* (`GM`) computes the match between recommended experts and the true answerers of test questions as: $\mathtt{GM} = \frac{100}{n} \sum_{k=1}^{n} \left(|\hat{q}_e^{(k)} \cap q_e^{(k)}| / |q_e^{(k)}| \right)$. Metric `GM` is equivalent to *Recall* in the information retrieval; *(2) Precision at N* (`PN`) measures the percentage of discovered experts that match with the actual answerers of each test question as: $\mathtt{PN} = \frac{1}{n} \sum_{k=1}^{n} |\hat{q}_e^{(k)} \cap q_e^{(k)}| / |\hat{q}_e^{(k)}|$; *(3) F1-score* (`F1`): $\mathtt{F1} = \frac{1}{n} \sum_{k=1}^{n} 2\mathtt{GM}_k \mathtt{PN}_k / \mathtt{GM}_k + \mathtt{PN}_k$, where $\mathtt{GM}_k$ and $\mathtt{PN}_k$ are metrics `GM` and `PN` computed for test question k. *(4) Matching Set Count* (`MSC`) indicates the percentage of test questions in which at least one of their recommended experts is among the true answerers: $\mathtt{MSC} = \frac{1}{n} \sum_{k=1}^{n} f(\hat{q}_e^{(k)}, q_e^{(k)})$, where $f(x, y) = 1$ if $x \cap y \neq \phi$; otherwise zero.

5.3 Experimental Setup

In each dataset, 90% of questions are used for training and 10% for testing. For each test question, all experts are potential answerers. Subgraphs g_q and g_e are created using respective first-order neighbors. RBF kernel parameters range from $\mu_1 = 1.0$ to $\mu_{11} = -0.9$, and $\sigma_1 = 10^{-3}$ with $\sigma = 0.1$ for kernels $\mathbf{K}^{(t)}$ and $\mathbf{K}^{(c)}$. Embedding dimensions are $d = 128$ and $d' = 300$. We utilized `Adam` [26] as the optimizer with an exponential decay learning rate initialized from $\{1 \times 10^{-5}, 1 \times 10^{-4}\}$ and other parameters with default values. For `EnC`, based on author explanations, it is implemented with two convolutional layers and an embedding dimension of 128. We used an MLP with two layers for both our method and `EnC`. In `CQR` [5], all experts' availabilities are equally weighed. For other baselines, we adhered to default parameters from their original implementations. The best results from five repetitions are reported.

5.4 Impact of Topological and Textual Encoding

We first conduct experiments to investigate the impact of end-to-end jointly learning of the topological and textual information extracted from the CQA environment. As such, three variations of the proposed technique are built as follows: **TextOnly**, employs

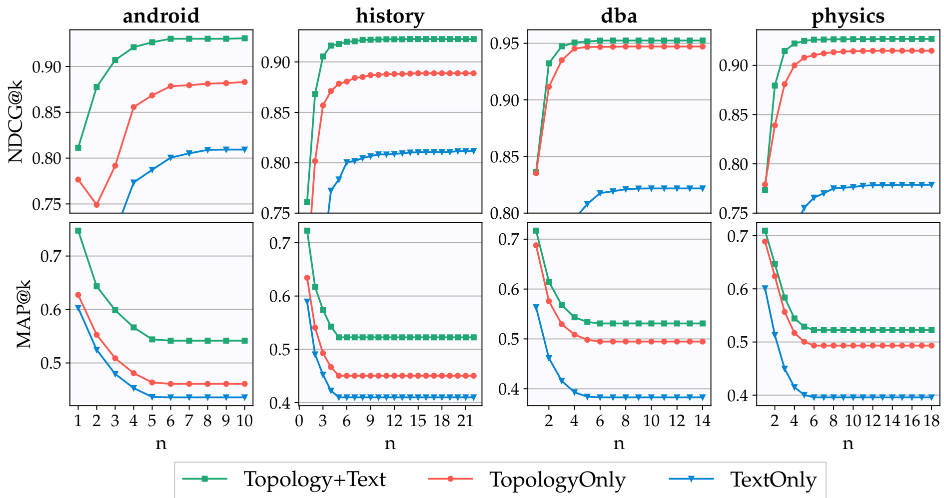

Figure 3: Impact of employing topological and textual information on predicting expert ranks based on ranking metrics NDCG and MAP at search depth n.

only the textual information, **TopologyOnly**, utilizes only topological information, and **Topology+Text**. uses both textual and topological information. The methods are used to rank the true answerers of test questions and random negative samples with the same number of true answerers. The ground truth ranks are obtained based on the answers' voting scores of the experts. The ranking metrics are employed to evaluate the results as shown in Fig. 3. In this figure, n denotes the rank depth determined based on the maximum number of answerers of test questions in each dataset. We observe that: 1) the superior expert ranks in terms of NDCG and MAP is achieved when the model utilizes both the topological and textual data; 2) the topological information is more informative for ranking experts compared to only employing textual information; 3) using both the topological and textual data improves the quality of experts' ranks on average by 3.4% in terms of NDCG and 11.29% in terms of MAP compared to the variant utilizing only the topological data on all datasets. Thus, we report the results of the variant employing both the topological and textual data as our best model in the rest of this paper.

5.5 Quality of Expert Ranks

We conduct comprehensive experiments to evaluate the effectiveness of our proposed approach against existing expert finding baselines. Our primary focus is on assessing the quality and accuracy of the predicted rankings of experts. In our experimental setup, each method ranks the individuals who have provided true answers to test questions, identified based on their voting scores from established experts in the domain. For a thorough evaluation, we augment each test question with an equal number of randomly selected negative samples, in addition to the true answerers.The results are depicted in Fig. 4. We make several observations: 1) Our model achieves consistently superior expert ranks in terms on both NDCG and MAP on all datasets. It indicates that our model better identifies and ranks the actual answerers of test questions compared the baselines. The improvement ratios are on average 4.4% and 4.6% in terms of NDCG and 7.1% and 9.3% in teams of MAP compared to our best baselines, namely NeR and EnC respectively; 2) The topological data extracted from the CQA environment is more informative for expert ranking compared to only using textual information. The experiments indicate that the learning to rank models which only utilize the textual similarities between the new question and past answers of experts obtain the worst results compared the other techniques in terms of ranking metrics.

Table 2: Comparison of the methods when they discover groups with the same size as the true answerers.

Datasets	GM/PN/F1(%)				MSC(%)			
	NeR	EnC	CQR	Our	NeR	EnC	CQR	Our
android	4.0	6.6	4.6	**35.1**	9.1	15.9	12.5	**63.6**
history	6.5	11.5	10.3	**69.2**	16.6	29.0	27.2	**91.7**
dba	6.5	8.7	11.0	**77.7**	13.8	20.3	25.2	**97.2**
physics	4.7	7.6	8.1	**64.3**	11.8	17.9	18.7	**89.6**

5.6 How Collaborative are Discovered Experts?

In this section, we perform experiments to assess the collaborative nature of the experts identified by our methods. To this end, all experts in each dataset are ranked by NeR, EnC the top two expert-finding performers from the previous experiment, and our method given a test question. Then, top τ experts are employed as a list of ranked experts to answer the question. Note that method CQR retrieves a collaborative ranked experts given a test question with a specific size. First, the methods are applied to find ranked experts with the same size as the true answerers of the test questions. The results are reported in Table 2. Since the size of predicted ranked lists of experts and the ground truth is the same, we have GM = PN = F1. The experiments reveal that our approach discovers more collaborative experts compared to the baselines. In other words, it discovers ranked experts with on average over 6.1 times superior results in terms of GM, PN, and F1 metrics and around 3.2

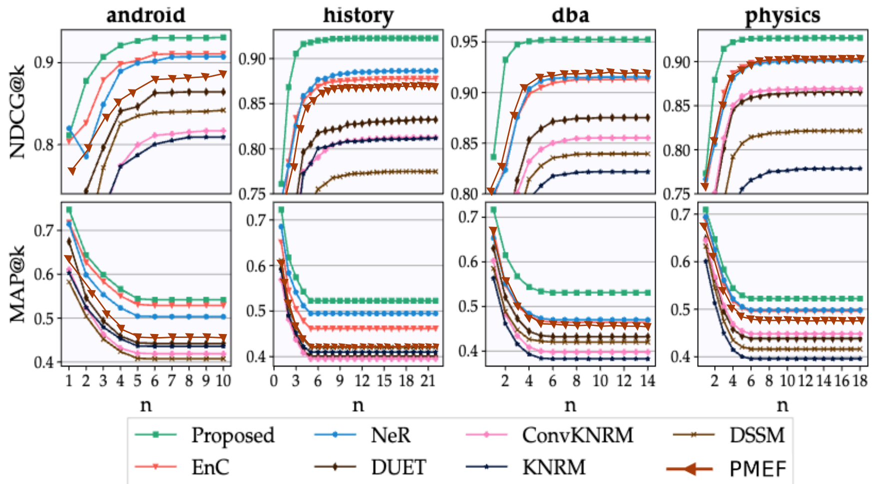

Figure 4: Comparison of the methods based on the quality of expert ranks using ranking metrics at search depth n.

times better results considering measure MSC compared to methods EnC and CQR.

5.7 Group Size Impact on Expert Collaboration Effectiveness

We investigate the impact of the size of groups on the collaboration among ranked experts obtained by each method. Thus, the parameters of the methods are set to discover expert lists with different sizes ranging between one and ten. The metrics are computed based on the discovered lists and reported in Fig 5. We summarize our findings as follows: 1) The proposed method consistently outperforms the baselines in terms of evaluation metrics GM, PN, F1, and MSC. Furthermore, CQR achieves the second place. 2) Our method recommends on average 68.6% of actual answerers of test questions as the members of new groups with sizes ranging from one to ten which it is roughly 3.3, 3.9, and 6.3 times improvement in terms of metric GM compared to baselines CQR, EnC, and NeR respectively. 3) On averages 39.5% of members of ranked expert lists discovered by our method are from the true answerers (2^{nd} row in the figure). That is roughly 4.1 times improvement compared to our best baseline CQR. 4) Our method recommends on average 3.8, 4.4, and 7.2 times superior ranked experts in terms of F1 score compared to baselines CQR, EnC, and NeR respectively. 5) In 83.9% of lists recommended by our method at least one of their members is among the true answerers. It is on average 3.7, 3.9, 6.4 time improvements compared to EnC, CQR, and NeR respectively (last row in the figure).

5.8 Comparison with Team Formation Techniques

We also compare our method against techniques for finding a team of experts (TE). As we do not have control over the size of groups formed by such methods, TE baselines are first applied on each test question and the size of a discovered group is used as an input for our method and other baselines namely CQR, EnC, and NeR. The results are reported in Tables 3, 4, and 5. The average size ($\overline{\tau}$) of ranked expert lists is shown for each dataset in the tables. Our observations are summarized as follows: 1) Our method consistently outperforms the baselines in terms of all the evaluation metrics. 2) The proposed model shows a significant superiority in terms of metrics GM, PN, and F1, with improvements of approximately 2.2 times compared to the best baseline CC as shown in Table 3. The improvement in the MSC metric is about 118% compared to our best baseline CC. 3) As reported in Table 4, our method demonstrates superior performance in terms of collaboration metrics compared to the baselines. The improvement is around 313% in terms of GM, which significantly exceeds improvements for other metrics compared to our best baseline CQR and second best baseline CO, respectively. 4) The experiments reported in Table 5 indicate that experts retrieved by our method more closely match the actual answerers compared to the baselines. The results reveal that our approach achieves on average 2.1 times superior results in terms of MSC compared to our best baseline CQR.

6 CONCLUSION AND FUTURE WORK

This paper addresses the problem of collaborative expert discovery on CQA platforms by introducing a novel end-to-end framework that simultaneously captures textual and topological information from the environment. Our proposed sub-graph encoding technique highlights the similarity among compact sub-graphs derived from the CQA network, while the semantic encoding component retains the textual congruence between a question and its answer. Through extensive experiments, we demonstrate the superior performance of our model in expert ranking quality. Furthermore, experts ranked by our method achieve higher collaboration scores compared to the state of the art. In our future work, we plan to investigate the dynamics of expert collaboration by examining how temporal factors and socio-psychological elements influence experts' effectiveness within Community Question Answering (CQA) platforms. We aim to analyze how the timing of interactions affects the quality and efficacy of collaborative responses and to explore the roles of reputation and authority in shaping such collaborations.

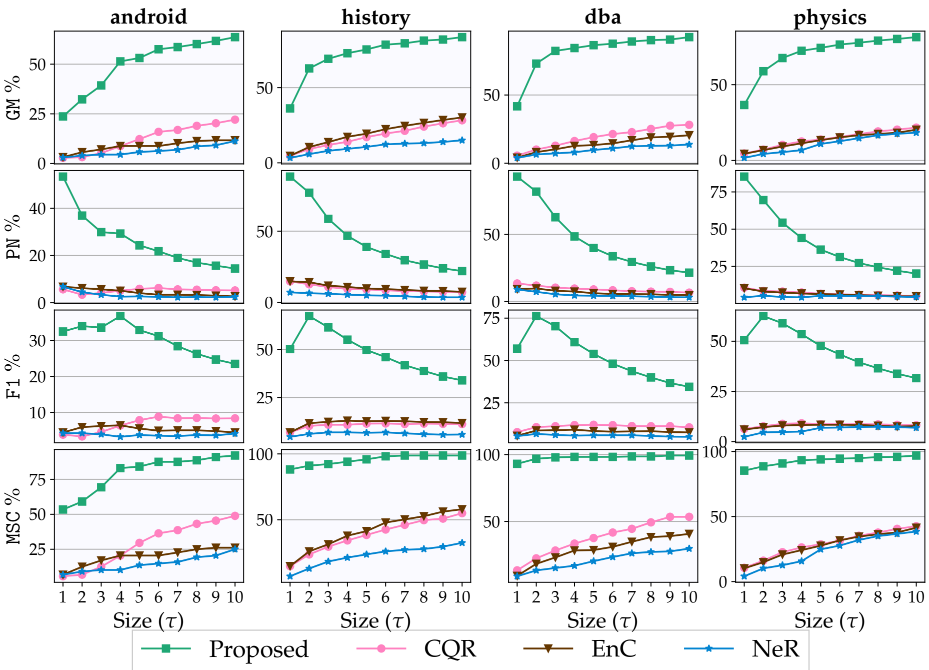

Figure 5: Comparison of the methods based on collaboration metrics.

Table 3: Performance Analysis of Top Expert Finding Methods (NeR, Enc) and Team Formation Methods (CQR, CC) against Our Method in Terms of Team Size as Determined by CC [31].

Datasets	$\bar{\tau}$	GM(%)					PN(%)					F1(%)					MSC(%)				
		CC	NeR	CQR	EnC	Our	CC	NeR	CQR	EnC	Our	CC	NeR	CQR	EnC	Our	CC	NeR	CQR	EnC	Our
android	3.5	15.6	4.2	8.6	7.6	**46.2**	10.0	3.0	6.1	5.5	**31.4**	11.9	3.3	6.9	6.1	**35.8**	34.9	9.6	20.5	18.1	**75.9**
history	3.0	14.4	7.7	11.5	13.5	**65.7**	14.0	6.3	12.0	13.6	**60.7**	13.1	6.5	10.8	12.6	**58.8**	37.3	17.8	29.0	30.2	**91.7**
dba	2.7	13.4	6.9	11.6	10.4	**73.1**	12.5	6.6	10.9	9.5	**67.7**	11.9	6.1	10.2	9.1	**66.0**	30.5	14.5	24.8	23.0	**95.4**
physics	2.9	9.5	5.7	9.6	9.1	**61.5**	9.5	4.6	8.9	8.7	**58.7**	8.6	4.7	8.5	8.1	**55.4**	22.9	13.1	20.2	19.8	**89.8**

Table 4: Performance Analysis of Top Expert Finding Methods (NeR, Enc) and Team Formation Methods (CQR, CC) against Our Method in Terms of Team Size as Determined by CO [24].

Datasets	$\bar{\tau}$	GM(%)					PN(%)					F1(%)					MSC(%)				
		CO	NeR	CQR	EnC	Our	CO	NeR	CQR	EnC	Our	CO	NeR	CQR	EnC	Our	CO	NeR	CQR	EnC	Our
android	5.1	17.9	5.2	12.9	8.2	**56.0**	8.4	2.9	5.8	3.9	**27.4**	11.0	3.6	7.8	5.2	**34.9**	38.6	12.0	30.1	19.3	**86.7**
history	4.9	15.4	9.8	15.3	18.2	**71.9**	9.7	5.2	9.9	11.2	**46.0**	11.2	6.6	11.2	13.0	**52.7**	37.3	21.9	36.1	40.8	**94.1**
dba	5.7	14.3	10.4	20.6	14.7	**86.6**	6.3	4.7	9.2	6.6	**38.0**	8.3	6.1	12.1	8.6	**50.6**	31.9	22.3	41.5	31.2	**97.9**
physics	6.4	11.0	13.3	16.0	15.9	**76.2**	4.7	4.7	6.4	6.3	**32.1**	6.2	6.7	8.6	8.5	**43.0**	25.5	28.0	32.7	33.1	**94.7**

Table 5: Performance Analysis of Top Expert Finding Methods (NeR, Enc) and Team Formation Methods (CQR, CC) against Our Method in Terms of Team Size as Determined by CS [25].

Datasets	$\bar{\tau}$	GM(%)					PN(%)					F1(%)					MSC(%)				
		CS	NeR	CQR	EnC	Our	CS	NeR	CQR	EnC	Our	CS	NeR	CQR	EnC	Our	CS	NeR	CQR	EnC	Our
android	7.6	20.3	8.5	18.3	10.6	**60.0**	6.6	2.4	5.7	3.4	**18.6**	9.8	3.7	8.5	5.1	**27.9**	43.4	19.3	42.2	24.1	**89.2**
history	7.1	19.8	12.1	21.4	23.9	**79.3**	7.3	4.3	7.8	8.5	**30.8**	10.3	6.1	11.0	12.1	**42.6**	45.6	26.0	46.2	49.7	**98.2**
dba	5.6	13.9	10.6	20.9	14.4	**87.1**	5.8	4.3	9.3	6.0	**38.1**	7.9	6.0	12.4	8.2	**51.4**	32.6	22.7	41.1	30.9	**98.2**
physics	9.6	9.7	17.6	20.9	20.4	**80.8**	2.6	4.3	5.3	5.1	**21.8**	4.0	6.8	8.2	8.0	**33.4**	22.2	37.3	41.2	41.2	**96.1**

REFERENCES

[1] Maddalena Amendola, Andrea Passarella, and Raffaele Perego. 2024. Towards Robust Expert Finding in Community Question Answering Platforms. (2024), 152–168.

[2] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. 2006. Formal models for expert finding in enterprise corpora. In *SIGIR*. ACM, 43–50.

[3] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. 2009. A language modeling framework for expert finding. *Inf. Process. Manag.* 45, 1 (2009), 1–19.

[4] Krisztian Balog, Yi Fang, Maarten de Rijke, Pavel Serdyukov, and Luo Si. 2012. Expertise Retrieval. *Found. Trends Inf. Retr.* 6, 2-3 (2012), 127–256.

[5] Shuo Chang and Aditya Pal. 2013. Routing questions for collaborative answering in community question answering. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*. 494–501.

[6] S. Chang and A. Pal. 2013. Routing questions for collaborative answering in Community Question Answering. In *ASONAM*. IEEE Computer Society, Los Alamitos, CA, USA, 494–501.

[7] X. Cheng, S. Zhu, S. Su, and G. Chen. 2017. A Multi-Objective Optimization Approach for Question Routing in Community Question Answering Services. *IEEE Transactions on Knowledge Data Engineering* 29, 09 (2017), 1779–1792.

[8] Xiang Cheng, Shuguang Zhu, Sen Su, and Gang Chen. 2018. A Multi-Objective Optimization Approach for Question Routing in Community Question Answering Services (Extended Abstract). In *ICDE*. IEEE Computer Society, 1765–1766.

[9] Jason H. D. Cho, Yanen Li, Roxana Girju, and Chengxiang Zhai. 2015. Recommending forum posts to designated experts. In *IEEE BigData*. IEEE Computer Society, 659–666.

[10] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *WSDM*. 126–134.

[11] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *WSDM*. 126–134.

[12] Arash Dargahi Nobari, Sajad Sotudeh Gharebagh, and Mahmood Neshati. 2017. Skill translation models in expert finding. In *SIGIR*. 1057–1060.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. Association for Computational Linguistics, 4171–4186.

[14] Roohollah Etemadi, Morteza Zihayat, Kuan Feng, Jason Adelman, and Ebrahim Bagheri. 2021. Collaborative Experts Discovery in Social Coding Platforms. In *CIKM*. 3009–3013.

[15] Roohollah Etemadi, Morteza Zihayat, Kuan Feng, Jason Adelman, and Ebrahim Bagheri. 2023. Embedding-based team formation for community question answering. *Inf. Sci.* 623 (2023), 671–692. https://doi.org/10.1016/j.ins.2022.09.036

[16] Jiafeng Guo, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2019. MatchZoo: A Learning, Practicing, and Developing System for Neural Text Matching. In *Proceedings of ACM SIGIR'19* (Paris, France) *(SIGIR)*. 1297–1300.

[17] Jinwen Guo, Shengliang Xu, Shenghua Bao, and Yong Yu. 2008. Tapping on the potential of q&a community by recommending answer providers. In *CIKM*. ACM, 921–930.

[18] Radin Hamidi Rad, Hossein Fani, Ebrahim Bagheri, Mehdi Kargar, Divesh Srivastava, and Jaroslaw Szlichta. 2023. A variational neural architecture for skill-based team formation. *ACM Transactions on Information Systems* 42, 1 (2023), 1–28.

[19] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. 2333–2338.

[20] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.

[21] Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. 2006. A framework to predict the quality of answers with non-textual features. In *SIGIR*. 228–235.

[22] Zongcheng Ji and Bin Wang. 2013. Learning to rank for question routing in community question answering. In *CIKM*. 2363–2368.

[23] Pawel Jurczyk and Eugene Agichtein. 2007. Discovering authorities in question answer communities by using link analysis. In *CIKM*. ACM, 919–922.

[24] Mehdi Kargar, Lukasz Golab, Divesh Srivastava, Jaroslaw Szlichta, and Morteza Zihayat. 2020. Effective Keyword Search over Weighted Graphs. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[25] Abeer Khan, Lukasz Golab, Mehdi Kargar, Jaroslaw Szlichta, and Morteza Zihayat. 2020. Compact group discovery in attributed graphs and social networks. *Information Processing & Management* 57, 2 (2020), 102054.

[26] D. Kingma and J. Lei Ba. 2015. A method for stochastic optimization. *International Conference on Learning Representations. San Dego: ICLR* (2015).

[27] Yue Kou, Derong Shen, Quinn Snell, Dong Li, Tiezheng Nie, Ge Yu, and Shuai Ma. 2020. Efficient Team Formation in Social Networks based on Constrained Pattern Graph. In *ICDE*. 889–900.

[28] Vaibhav Krishna, Vaiva Vasiliauskaite, and Nino Antulov-Fantulin. 2022. Question routing via activity-weighted modularity-enhanced factorization. *Soc. Netw. Anal. Min.* 12, 1 (2022), 155.

[29] Dipankar Kundu and Deba Prasad Mandal. 2019. Formulation of a hybrid expertise retrieval system in community question answering services. *Appl. Intell.* 49, 2 (2019), 463–477.

[30] Dipankar Kundu, Rajat Kumar Pal, and Deba Prasad Mandal. 2021. Topic sensitive hybrid expertise retrieval system in community question answering services. *Knowl. Based Syst.* 211 (2021), 106535.

[31] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *SIGKDD*. 467–476.

[32] Baichuan Li and Irwin King. 2010. Routing questions to appropriate answerers in community question answering services. In *CIKM*. ACM, 1585–1588.

[33] Baichuan Li, Irwin King, and Michael R. Lyu. 2011. Question routing in community question answering: putting category in its place. In *CIKM*. ACM, 2041–2044.

[34] Ming Li, Ying Li, Yueyun Chen, and Yingcheng Xu. 2021. Batch recommendation of experts to questions in community-based question-answering with a sailfish optimizer. *Expert Syst. Appl.* 169 (2021), 114484.

[35] Zeyu Li, Jyun-Yu Jiang, Yizhou Sun, and Wei Wang. 2019. Personalized question routing via heterogeneous network embedding. In *AAAI*, Vol. 33. 192–199.

[36] Duen-Ren Liu, Yu-Hsuan Chen, Wei-Chen Kao, and Hsiu-Wen Wang. 2013. Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Information Processing & Management* 49, 1 (2013), 312 – 329.

[37] Hongtao Liu, Zhepeng Lv, Qing Yang, Dongliang Xu, and Qiyao Peng. 2022. Efficient Non-sampling Expert Finding. In *CIKM*. 4239–4243.

[38] Hongtao Liu, Zhepeng Lv, Qing Yang, Dongliang Xu, and Qiyao Peng. 2022. ExpertBert: Pretraining Expert Finding. In *CIKM*. 4244–4248.

[39] Xiaoyong Liu, W Bruce Croft, and Matthew Koll. 2005. Finding experts in community-based question-answering services. In *CIKM*. 315–316.

[40] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. 2011. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI'11*. 2857–2866.

[41] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *WWW*. 1291–1299.

[42] Saeedeh Momtazi and Felix Naumann. 2013. Topic modeling for expert finding using latent Dirichlet allocation. *WIREs Data Mining Knowl. Discov.* 3, 5 (2013), 346–353.

[43] Narjes Nikzad-Khasmakhi, Mohammad Ali Balafar, Mohammad-Reza Feizi-Derakhshi, and Cina Motamed. 2021. ExEm: Expert embedding using dominating set theory with deep learning approaches. *Expert Syst. Appl.* 177 (2021), 114913.

[44] Qiyao Peng and Hongtao Liu. 2022. ExpertPLM: Pre-training Expert Representation for Expert Finding. In *EMNLP*. 1043–1052.

[45] Qiyao Peng, Hongtao Liu, Zhepeng Lv, Qing Yang, and Wenjun Wang. 2023. Contrastive Pre-training for Personalized Expert Finding. In *EMNLP*. 15797–15806.

[46] Qiyao Peng, Hongtao Liu, Yinghui Wang, Hongyan Xu, Pengfei Jiao, Minglai Shao, and Wenjun Wang. 2022. Towards a Multi-View Attentive Matching for Personalized Expert Finding. In *The ACM Web Conference*. ACM, 2131–2140.

[47] Qiyao Peng, Hongtao Liu, Hongyan Xu, Yinghui Wang, and Wenjun Wang. 2023. PEPT: Expert Finding Meets Personalized Pre-training. *CoRR* abs/2312.12162 (2023).

[48] Qiyao Peng, Wenjun Wang, Hongtao Liu, Yinghui Wang, Hongyan Xu, and Minglai Shao. 2022. Towards comprehensive expert finding with a hierarchical matching network. *Knowl. Based Syst.* 257 (2022), 109933.

[49] Desislava Petkova and W. Bruce Croft. 2008. Hierarchical Language Models for Expert Finding in Enterprise Corpora. *Int. J. Artif. Intell. Tools* 17, 1 (2008), 5–18.

[50] Lingfei Qian, Jian Wang, Hongfei Lin, Bo Xu, and Liang Yang. 2022. Heterogeneous information network embedding based on multiperspective metapath for question routing. *Knowl. Based Syst.* 240 (2022), 107842.

[51] Radin Hamidi Rad, Hossein Fani, Mehdi Kargar, Jaroslaw Szlichta, and Ebrahim Bagheri. 2020. Learning to Form Skill-based Teams of Experts. In *CIKM*.

[52] Habibur Rahman, Senjuti Basu Roy, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. 2019. Optimized group formation for solving collaborative tasks. *The VLDB Journal* 28 (2019), 1–23.

[53] Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei, and Evangelos Milios. 2012. Finding expert users in community question answering. In *WWW*. 791–798.

[54] Fatemeh Riahi, Zainab Zolaktaf, M. Mahdi Shafiei, and Evangelos E. Milios. 2012. Finding expert users in community question answering. In *WWW*. ACM, 791–798.

[55] Anna Sapienza, Palash Goyal, and Emilio Ferrara. 2019. Deep Neural Networks for Optimal Team Composition. *Frontiers Big Data* 2 (2019), 14.

[56] Mohsen Shahriari, Sathvik Parekodi, and Ralf Klamma. 2015. Community-aware ranking algorithms for expert identification in question-answer forums. In *I-KNOW*. 8:1–8:8.

[57] Soroosh Sorkhani, Roohollah Etemadi, Amin Bigdeli, Morteza Zihayat, and Ebrahim Bagheri. 2022. Feature-based question routing in community question answering platforms. *Inf. Sci.* 608 (2022), 696–717.

[58] Jiankai Sun, Abhinav Vishnu, Aniket Chakrabarti, Charles Siegel, and Srinivasan Parthasarathy. 2018. Coldroute: effective routing of cold questions in stack exchange sites. *Data mining and knowledge discovery* 32, 5 (2018), 1339–1367.

[59] Jiankai Sun, Jie Zhao, Huan Sun, and Srinivasan Parthasarathy. 2020. EndCold: An End-to-End Framework for Cold Question Routing in Community Question

Answering Services. In *IJCAI*. 3244–3250.
[60] Andrew Turpin and Falk Scholer. 2006. User performance versus precision measures for simple search tasks. In *Proceedings of ACM SIGIR'06*. 11–18.
[61] Yijing Wang, Dachuan Xu, Donglei Du, and Ran Ma. 2020. Bicriteria algorithms to balance coverage and cost in team formation under online model. *Theoretical Computer Science* (2020).
[62] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *SIGIR*. 55–64.
[63] Fei Xu, Zongcheng Ji, and Bin Wang. 2012. Dual role model for question recommendation in community question answering. In *SIGIR*. ACM, 771–780.
[64] Baoguo Yang and Suresh Manandhar. 2014. Tag-based expert recommendation in community question answering. In *ASONAM*. IEEE Computer Society, 960–963.
[65] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. 2013. CQArank: jointly model topics and expertise in community question answering. In *CIKM*. ACM, 99–108.
[66] Sha Yuan, Yu Zhang, Wendy Hall, and Juan Bautista Cabotà. 2020. Expert finding in community question answering: a review. *AI Review* 53, 2 (2020), 843–874.
[67] Jun Zhang, Mark S. Ackerman, and Lada A. Adamic. 2007. Expertise networks in online communities: structure and algorithms. In *WWW*. ACM, 221–230.
[68] Xuchao Zhang, Wei Cheng, Bo Zong, Yuncong Chen, Jianwu Xu, Ding Li, and Haifeng Chen. 2020. Temporal Context-Aware Representation Learning for Question Routing. In *WSDM*. 753–761.
[69] Zhou Zhao, Lijun Zhang, Xiaofei He, and Wilfred Ng. 2014. Expert finding for question answering via graph regularized matrix completion. *IEEE Transactions on Knowledge and Data Engineering* 27, 4 (2014), 993–1004.
[70] Xiaolin Zheng, Zhongkai Hu, Aiwu Xu, Deren Chen, Kuang Liu, and Bo Li. 2012. Algorithm for recommending answer providers in community-based question answering. *J. Inf. Sci.* 38, 1 (2012), 3–14.
[71] Guangyou Zhou, Kang Liu, and Jun Zhao. 2012. Monolingual-Based Translation Model for Question Routing. In *Pattern Recognition - Chinese Conference, CCPR (Communications in Computer and Information Science, Vol. 321)*. Springer, 630–637.
[72] Hengshu Zhu, Enhong Chen, Hui Xiong, Huanhuan Cao, and Jilei Tian. 2014. Ranking user authority with relevant knowledge categories for expert finding. *World Wide Web* 17, 5 (2014), 1081–1107.