Information Sciences 623 (2023) 671–692

Contents lists available at ScienceDirect

Information Sciences

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

Embedding-based team formation for community question answering

Roohollah Etemadi [a], Morteza Zihayat [a], Kuan Feng [b], Jason Adelman [b], Ebrahim Bagheri [a,*]

[a] Ryerson University, Canada
[b] IBM Canada Ltd., Canada

ARTICLE INFO

Article history:
Received 8 January 2022
Received in revised form 7 September 2022
Accepted 9 September 2022
Available online 17 October 2022

Keywords:
Team formation
Network embedding
Learn to ranking
Skill coverage

ABSTRACT

Finding a qualified individual who can independently answer a question on a community question answering platform is becoming more challenging due to the increasing multidisciplinary nature of posted questions. As such, finding a group of experts to collaboratively answer the questions is of paramount importance. To this end, we propose a novel approach to form teams of experts who can collectively answer new questions. The proposed approach, called `team2box`, learns neural embedding representations based on the content of the posted questions, experts' engagement with these questions, and past expert collaboration history in order to form a team to answer the posted question. It embeds experts and questions as points and existing teams as regions within the embedding space. Therefore, `team2box` forms a team whose members (1) collectively cover the knowledge required to answer a question, (2) have successful past experience in jointly answering similar questions, and (3) can work efficiently together to answer the question. Extensive experiments on real-life datasets from Stack Exchange show that `team2box` outperforms the state-of-the-art by discovering teams with on average 38.97% more covering the skills required to answer new questions and employing experts with collectively a high expertise level.

1. Introduction

Community-based question answering (CQA) systems such as *Stack Exchange*, and *Quora* are popular web-based services that match information seeking users with their knowledgeable counterparts. In such context, the task of matching appropriate users that can effectively answer questions posted by information seekers is important and often known as *expert finding* or *question routing* [1–4]. Existing works on expert finding primarily focus on retrieving a ranked list of experts that are relevant to a posted question. This is limited from three main perspectives:

1. (ℓ1) An exploration of questions on collaborative question answering platforms shows that questions are often complicated and multidisciplinary. The plots in Fig. 1 show different metrics representing the multidisciplinary nature of questions in the Stack Exchange dataset used and introduced later in this paper. The figure shows distribution of the number of answers (left chart), number of question tags (middle chart), and the average dissimilarity between pairs of tags

* Corresponding author.
E-mail addresses: etemadir@ryerson.ca (R. Etemadi), mzihayat@ryerson.ca (M. Zihayat), kuan@ca.ibm.com (K. Feng), jadelman@ca.ibm.com (J. Adelman), bagheri@ryerson.ca (E. Bagheri).

https://doi.org/10.1016/j.ins.2022.09.036
0020-0255/

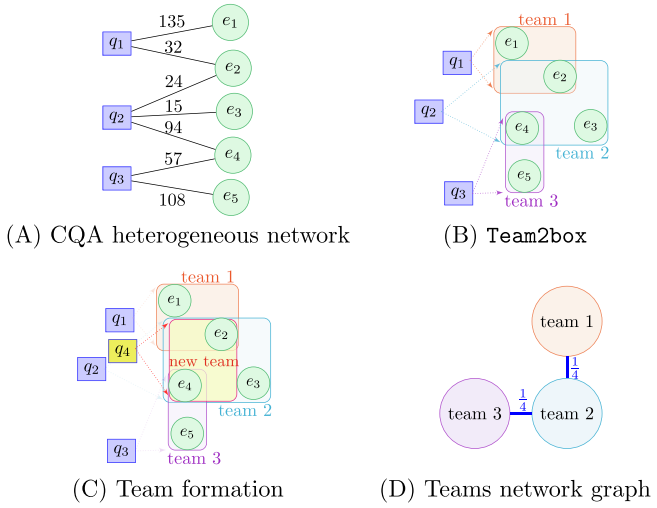

Fig. 1. Distribution of number of answers (left plot) and tags for each question (middle plot), as well as the average dissimilarity between tags assigned to questions and tags mentioned in questions answered by each expert (right plot).

assigned to questions versus the average dissimilarity between tags assigned to questions answered by authors (right chart). As shown, more than 37% of questions have more than two answers (the left chart), which implies the need for collaborations to come up with acceptable answers. Furthermore, around 60% of questions have more than two tags. The higher number of tags for each question can be viewed as a sign for the need for broader knowledge or expertise to answer a question. Since one could argue that tags might all be describing the same topic, we also visualize the dissimilarity between the tags of the same question in the right plot. From the right plot, a lower average dissimilarity between the pairs of tags[1] shows a higher homogeneity between the tags. As illustrated, the average dissimilarity between the pairs of tags assigned to questions answered by a user is much lower compared to the average dissimilarity between the tags of each question. This means that questions cover a wider range of tags compared to experts, indicating that, questions require the involvement of more than one expert to cover all aspects of the question.

2. ($\ell 2$) While existing methods on question routing are designed to recommend individual experts for each question, it is possible to take the top-k recommended experts from these methods and build a team who would collaboratively work on each question. However, given existing question routing methods have not been designed to consider complementarity of skill sets in the top-k retrieved experts, the highly ranked experts may end up being those that have the same set of skills. Therefore, creating teams that have members with identical or highly overlapping skill sets would not be appropriate as it would not add much benefit over just selecting a single expert; and
3. ($\ell 3$) Last but not least, even if the top-k experts retrieved by the question routing methods have complementary skill sets, they must additionally satisfy at least one requirement relating to their likelihood of successful collaboration. In other words, a successful team is one whose members have complementary skill sets and the members are able to effectively work together as a team. As such, the top-k experts retrieved by a question routing method might not form an ideal collaborative team even if they have the right set of skills. We hypothesize that answering complicated questions requires collaboration among the answerers.

For these reasons, inspired by [5], we employ the idea of *team formation* to expand the task of expert finding to one of developing teams of experts who can collaboratively address an information need. The problem we address has some resemblance to the problem of team formation in social networks [6], which is primarily based on selecting a sub-graph from a larger social network where a set of required skills are covered by the selected user nodes. It has been shown that finding an optimal solution to this problem is NP-hard and therefore, existing works adopt heuristic methods to identify appropriate teams [7–9]. The shortcoming of such graph-based techniques is that given the computational complexity of the problem, they often only explore restricted parts of the graph and therefore result in *sub-optimal* teams.

In light of the limitations of expert finding methods and team formation techniques, we propose a neural embedding based approach, called `team2box`, to address the task of collaborative team formation for community question answering. The major distinguishing aspects of our work can be enumerated as: (a) Unlike team formation techniques [7,9,10], `team2box` does not restrict the search space as it fully embeds the network through a specialized neural embedding technique. `team2box` first maps existing questions, answerers, existing teams (subgraphs of CQA network graph), and new questions

[1] We compute tag embedding dissimilarity as the euclidean distance between the embedding vectors of pairs of tags based on a word2vec model trained over the Stack Exchange corpus.

into the same vector space. Then, it searches for closest teams to the new question in the vector space. Such a search technique allows team2box to explore the entire CQA network graph. In contrast, team formation techniques adopt a heuristic strategy and start from potential answerers (nodes in the CQA network graph) with a high chance of being selected as members of a new team. Then, they explore only the neighborhood of such nodes to find a compact subgraph to satisfy team formation constraints. (b) Moreover, our method moves beyond existing neural graph embedding techniques as it is able to not only embed the nodes of the heterogeneous graph but also its selected subgraphs. This allows team2box to embed experts and questions as points and existing teams (as selected subgraphs of the CQA network graph) as regions within the embedding space. This way, our method is able to place experts, questions and teams in the same embedding space; hence, facilitate the retrieval of an effective team for an incoming question. Our main contributions in this paper are summarized as follows:

- (1) We propose a novel neural embedding model to embed existing teams as regions into a vector space and used such latent representations to build new teams for newly posted questions. Each existing team is represented by a center vector and an offset in the embedding space. The center vector provides a representation for the embedding of the team members; while the offset shows the boundary of the team in the vector space. Teams with common members are embedded close to each other with overlapping boxes.
- (2) We conduct extensive experiments on five real-life datasets obtained from Stack Exchange, which is a popular community question answering platform to study the effectiveness of proposed approach. The performance of the proposed method is compared with expert finding and team formation baselines. The experiments reveal that our method is dataset independent and shows a consistent superior performance over the baselines in terms of the evaluation metrics.

The rest of this paper is organized as follows. We summarize existing works in Section 2. Then, we formulate the problem and propose our method in Sections 3 and 4, respectively. Section 5 contains our experimental results, the performance comparison of the methods, and our observations and findings. Our concluding remarks and future works are presented in Section 6.

2. Related work

We note that the field of Community Question Answering has received attention from different perspectives such as answer selection, determining unclear questions, question tagging, and predicting answer votes, to name a few. However, relevant work to team2box can be classified into two categories: 1) *expert finding* or *question routing* techniques, and 2) *team formation* approaches. The former techniques aim to retrieve an individual who has the highest required skill coverage and expertise level, while the latter focuses on the past collaboration of experts and finds a group of experts who collectively cover all the required skills and maximizes some objective functions such as expertise level, expert authority, and past collaboration.

2.1. Expert finding

Given a new question, a number of *expert finding* techniques retrieve experts with a high similarity between the content of their past questions and answers, and the textual content of the new question. To do so, different techniques such as language models, feature-based learn to rank model, hybrid approach integrating expert profile, reputation and link analysis, supervised learning, statistical and probabilistic topic modeling, translation models, and factorization machines have been used. Beside the textual information, the link structure among the entities in the CQA environment, i.e. questions, answers, askers, answerers, etc, has been utilized for expert finding [11]. More recent methods leverage experts' social network considering the fact that friends in a social network tend to have similar beliefs, values and interests (homophily). Such a social network can be an external network such as Twitter, Facebook, and LinkedIn [12], or an internal one built using the interactions between the experts in the community [1,2]. Table 1 compares some of such existing methods and our proposed technique based on the types of information they utilize.

2.1.1. Methods using textual data

These methods leverage textual data extracted from questions and their answers along with different metrics such as expertise score of experts, question tags, and expert and question specific statistical features to route new questions to proper experts. Liu et al. consider the expert finding task as an information retrieval problem [16]. To this end, a new question is considered as a new query and experts' profiles are viewed as documents. Then, given the new question, the profiles are ranked using information retrieval techniques. The experts with top-ranked profiles are more likely to be potential answerers for the new question. In this context, the content of past questions and answers of experts are used to build their profiles [16]. The work was improved in [17] by considering the availability scores of experts in the ranking of their profiles. Li et al. [18] used question categories to better estimate the expertise score of experts compared to the language models used in [16,17]. Previous language model based techniques suffer from the word mismatch problem between the content of new questions and the answerer profiles when the number of answers and questions of experts is small. Thus, Zhou et al. pro-

Table 1
Summary of some related existing expert finding methods.

Methods	Textual information		Structural information				
	question	answer	questions	tags	askers	answerers	external data
GRMC [13]	✓	×	×	×	×	✓	✓
HSNL[14]	✓	✓	✓	×	×	✓	✓
RMNL [15]	✓	×	✓	×	×	✓	✓
AMRNL [12]	✓	✓	✓	×	×	✓	✓
NeRank [1]	✓	×	×	×	✓	✓	×
Seq [2]	×	×	✓	✓	✓	✓	×
Our approach	✓	✓	✓	×	×	✓	×

posed a joint relevance and answer quality learning technique to address the problem [19]. Furthermore, to better perform question routing, question-specific and user-specific statistical features have been employed in [20] to train a learn to rank model for expert finding. The authors in [21] used the similarities between the content of a new question and the experts' profiles, the question topics and the topics of experts' interests, and structural information between the asker and experts in the information network. They found that content and topics of interest are more robust metrics for expert finding compared to similarity in an information network.

2.1.2. Methods using textual and network data

The network data extracted from interactions among entities in a CQA platform has been used along with textual data to boost the quality of expert ranks given a new question. Zhou et al. [11] used both the link structure and the topical similarity between askers and answerers. To this end, a directed graph was built using askers and ansewerers to model question–answer relationships between askers and answerers. Then, a PageRank-based method was used to find experts. This work has been improved by Liu et al. using link analysis techniques and topical similarities between users and questions [22]. The authors in [23] utilize a directed heterogeneous user graph to obtain the expertise level of users on different topics, and to produce a ranked list of users based on their interests and expertise for each topic. Such ranked lists are utilized to rank experts and existing answers, and find similar questions given a new question [23]. Later, Yang et al. found that question tags tend to be more informative than user profiles or user topics obtained from the content of questions and answers which are ignored in most of previous work [24]. Another approach to perform expert finding is to view it in the form of a missing value estimation problem [13]. This approach chooses experts for answering the new questions based on a rating matrix which shows the quality of the answerers based on the answers they have provided for past questions. The rating scores of experts are known for the questions they have answered. Thus, missing values in the rating matrix needs to be predicted and hence expose the suitability of a user for a given question. The social relationships of experts from external social networks can be used to enrich the prediction of missing values in the rating matrix. The authors in [14,15,12] model this problem as a heterogeneous social network graph whose nodes consist of entities such as questions, answers and answerers. The relationships between the answerers, as well as the question–answer and answer-answerer pairs are used to constitute the graph edges. In a similar work, while using a heterogeneous graphical representation, Li et al. [1] embed question content, their askers, and answerers into latent space. The learned latent representations of the three entities are used as an input to a convolutional scoring function to rank existing answerers for a new question. Similarly, Sun et al. [2] construct a heterogeneous network using questions, tags, askers, and answerers. Then, a graph convolutional network is employed to learn the embeddings of the four types of entities. Given a new question, these embeddings are fed into a feed-forward neural network to predict the score of an answer provided by each answerer. The predicted scores are used to rank existing answerers given a new question. Akin to [1,2], our proposed `team2box` approach utilizes both the textual data and the structural information of a social network constructed using experts' relationships when answering common questions. Unlike earlier work [1,2], `team2box` does not overlook answer content and considers them when learning embedding representations. Our experiments show that using such important information results in finding experts with a higher likelihood of having the required skill sets to answer a new question. Furthermore, different from those in [1,2], `team2box` utilizes the voting scores of answers to form a weighted CQA network. Such an approach allows `team2box` to learn effective latent representations of questions and experts, and results in retrieving experts with the highest levels of expertise.

2.1.3. Other approaches

In many cases, the features extracted from existing questions and answers are treated equally for ranking experts. However, in some cases some features in a field are more significant than others [25]. Therefore, a hierarchical attention factorization machines based method has been designed to assign attention weights to each individual feature, and also represent the importance of each pairwise interaction among features [26]. Following such an idea, Kundu et al. [27] extract the topics of existing questions using the LDA method, and utilize the distribution of answers of experts on such topics along with their quality to estimate the expertise, reputation and authority of experts on each topic. Such signatures for experts are used to estimate their ranking scores given the distributions of the LDA topics on a new question.

Most expert finding methods have tended to employ the textual and structural information among existing questions to determine the experts' levels of expertise given an new question. A problem arises when the new question is unlike any questions previously asked in the system which is called a cold question. In such a scenario, it is challenging to properly rank experts given a cold question. Fu et al. [28] have approached the problem by employing recurrent memory reasoning networks to focus on different parts of existing questions, and accordingly extracts information from the past contributions of experts.

Another interesting direction in the literature is to consider the system as communities of experts and route new questions into related communities which is called community question routing [29]. In such techniques, a new question should be routed into a community to maximize the quality of the answers provided by the members of the community to the question. One of the challenging tasks for such a problem is the task of modeling community features.

In other work, the impact of different metrics on the performance of CQA systems has been explored. For example, the authors of [30] have investigated the impact of contact rate and expertise differences on the effectiveness of synchronous CQA systems. They have found that participants (question askers and answerers) consider CQA systems with a low contact rate to be more useful. They found that in CQA systems, it may be useful to find answerers with slightly broader knowledge than the asker, and reserve the most expert answerers for the most knowledgeable askers [30]. Prior studies have found that a majority of the content on CQA sites is generated by a small group of users [31]. Thus, to increase the size of such a group, Le and Shah [31] have designed an approach to identify such potential experts early on to help a CQA platform provide support to these potential users to encourage and cultivate their activity in the system. Nie et al. proposed an end-to-end deep interactive embedding model to automatically label questions with tags in CQA platforms [32]. The proposed model uses two parallel deep models to compute the embedding representations of questions and topic tags. These embeddings are then regularized such that they can alleviate the problem of imbalanced topic distribution. More recently, Liu et al. utilized user interest drift and user quality to avoid routing questions to experts whit low willingness or potential to provide qualified answers [33]. Voting information of community users has also utilized to determine their expertise [34].

2.2. Team formation

More recently, graph-based search techniques have been used for finding a group of experts from expert networks [9,8,7]. Considering an expert network as a graph whose nodes are experts and edges represent the past collaboration of experts, the problem of team formation is to find a subgraph (experts) whose nodes cover a given set of skills while maximizing some objective functions. For instance, Chang and Pal have designed a greedy algorithm for the collaborative team formation for CQA [5]. They employ several heuristics to extract expertise of experts, their availability, and the compatibility of experts with each other in a CQA environment. To form a team of size n given a new question, this work first adds an expert with high expertise and availability degree into the team. Then, it computes the compatibility of each unselected expert with the members of the team. A new member is added into the team with higher expertise and compatibility. This process is repeated $n-1$ times to develop a team with size n.

For another example, in [35], the authors argue that the expertise level and the compactness of the team are important in team formation. In [8], the authors consider Constrained Pattern Graph during team discovery. They argue that by considering structure constraints and communication constraints on team members, the discovered teams are able to meet user requirements effectively. Nikolakaki et al. [36] argue that having clear roles for team members, and being mutually respected by their teammates for the assigned roles on the team improves a team's performance. In a recent study [37], the authors propose a team formation algorithm to minimize the communication cost and also minimize the cost between the team members and team leaders. In [38], the problem of team formation is aggregated with influence maximization to help organizations with social events. The workload of experts along with skill coverage and communication costs are factors that are considered to be influential for forming efficient teams [39]. These factors fairly allocate experts into teams in which none of the experts should be overloaded with tasks or being unfairly under-employed.

More recently, Khan et al. [7] proposed approximation algorithms to form compact teams with desired sizes in attributed graphs and social networks. The team members are chosen in a way that they are closely connected and each one possesses as many required skills as possible. Kargar et al. [9] studied the problem of group discovery over weighted node-labeled network graphs by considering ranking objectives that take edge distance and node costs into account. For successful collaboration of group members, two collaboration-related factors called *affinity* and *upper critical mass* are adapted from organizational science and social theories in [40]. The former represents the comfort-level between team members who work on the same task, while the latter is a constraint on the team size beyond which the collaboration effectiveness diminishes. In [41], the authors argue that real-world teams often have complex structures and deep hierarchies, and each team member occupies a distinct role in these structures. As such, the template of the new team is built using a graph in which each node in the graph is assigned a role. Then, they discover experts that can occupy the roles in the template graph, while minimizing the communication cost along the edges of the template graph. The concept of *team faultlines* has been adapted in [42]. Faultlines are hypothetical partitioning lines that divide a team into subgroups based on single or multiple attributes. Faultlines have been found to influence a team's performance in which subgroups caused by faultlines in a team possess a risk of costly conflicts and poor communication.

As mentioned earlier, the main limitation of existing team formation methods is that they rely on heuristic-based graph search that is (i) computationally expensive, and (ii) can result in sub-optimal teams due to how subgraphs are explored locally.

2.3. Contributions of this work

A number of expert finding methods have been developed over the last few years. However, finding a qualified expert who can independently answer a new question is becoming more challenging due to the increasing multidisciplinary nature of the posted questions. Similar to Chang and Pal's work [5], which we refer to as `RQC`, our approach, i.e., `team2box`, overcomes such an issue by expanding the task of expert finding to one of retrieving a group of experts that can collaboratively answer new questions. In contrast to `RQC`, our method does not need to extract expertise and compatibility of experts using different predefined static rules. Instead, our model learns the rules explicitly from the data collected from existing successful teams. Both our technique and `RQC` employ a graph structure but with different structures. `RQC` models the collaboration of experts as a co-occurrence graph. The nodes of such a graph denote experts and the edge between two nodes indicates corresponding experts for the nodes answered at least one common question. Then such a graph is translated into several forms as Similarity Graph, Distance Graph, Shortest Distance Graph, and Homophily Graph. In contrast, we represent questions, answerers, and their relationships as a heterogeneous network graph, and then our team2box method preserves the properties of the network in the embedding space using the proposed neural based embedding technique. Our method retrieves a team of experts in which team members collectively cover the skills required to answer a question while enjoying a high likelihood of positive collaboration among them by taking past team collaborations into account. To this end, `team2box` models the CQA system as a heterogeneous network. Akin to network embedding techniques designed for a variety of network types from simple graphs to heterogeneous ones [43], `team2box` learns the latent representation of entities in the CQA heterogeneous network. Our method moves beyond such network embedding techniques, and embeds selected subgraphs (existing teams) of the CQA network as regions in the vector space and its nodes as points in the same space.

Compared to existing team formation methods, `team2box` overcomes their limitations by (a) mapping the entire network graph into vector space, and (b) preserving the structure of existing teams as subgraphs of the network in the same vector space. Given a new question, our approach explores the entire network in the embedding space to retrieve closest subgraphs to the embedding of the question. Then, the members of the new team are chosen from the discovered subgraphs (teams).

3. Problem formulation

Let $Q = \{q_1, q_2, \ldots, q_n\}$ be a set of n questions, $A_i = \{a_1, a_2, \ldots, a_{n_i}\}$ be a set of n_i answers to question q_i, and $S_i = \{s_1, s_2, \ldots, s_{n_i}\}$ be a set of quality (voting) scores for the answers in A_i; where s_j is an integer calculated based on the difference between answer a_j's up-votes and down-votes which is assigned by users who viewed the answer. Also, allow $Tg_i = \{t_1, t_2, .., t_{z_i}\}$ to be a set of tags for question q_i assigned by its asker. Moreover, we let $E = \{e_1, e_2, \ldots, e_m\}$ to be a set of m experts (answerers). Terms *expert* and *answerer* are used interchangeably in the rest of the paper.

Definition: A **Community Question Answering (CQA) Heterogeneous Network** is denoted as $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{W})$ where $\mathcal{V}$ is a set of nodes, $\mathcal{E}$ a set of edges, and $\mathcal{T}$ is the set of node and edge types. Furthermore, let $\mathcal{T}_{\mathcal{V}}(\subset \mathcal{T})$ denote node types, which can be *questions (q)*, or *experts (e)*. Similarly, $\mathcal{T}_{\mathcal{E}}(\subset \mathcal{T})$ indicates a set of edge types which can be *question-expert (q—e)* relationships. Moreover, $\mathcal{W} : \mathcal{E} \rightarrow \mathcal{E}_w$ is a function mapping each edge to a weight, and $\mathcal{E}_w$ is the set of possible weights. Note that $\mathcal{E}_w$ is the voting scores (the difference between the up-votes and down-votes) of answers for (q—e) relationships.

Panel (A) in Fig. 2 shows a toy example of the CQA heterogeneous network with three questions (q_1, q_2, q_3), and five experts (i.e., e_1, e_2, e_3, e_4, e_5). The weight of edges shows the voting scores of the answers provided by experts. For example, the weight of edge q_1—e_1 indicates that the difference between the up- and down-votes of the answer provided by e_1 to question q_1 is 135.

Problem: **Team Formation in Community Question Answering.** Given a CQA Heterogeneous Network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{W})$ and a new question q_{new} with c words and the tag set $tg^{(q_{new})}$, the task aims to find a team of experts denoted by T with size τ that can answer q_{new} in which the members of a discovered team T should:

1. (O1) collectively cover the required skills to answer the question.
2. (O2) have a high level of expertise in the required skills.
3. (O3) exhibit willingness to work together as a team.

A naive embedding-based solution for this team formation problem is to learn node embeddings for each $v_i \in \mathcal{V}$. Then, top-τ experts representing answers to the most similar questions to q_{new} would be selected as the team. We argue that this naive approach is not effective for three main reasons: 1) the members of the formed team might not possess the skills required to answer q_{new}; 2) the team members might not have already *successfully provided* high quality answers to the existing questions that are similar to the new one; and more importantly, 3) the team members might not be able to work together efficiently, e.g., lack of *past contributions* to similar shared questions, in order to successfully collaborate to answer the new question.

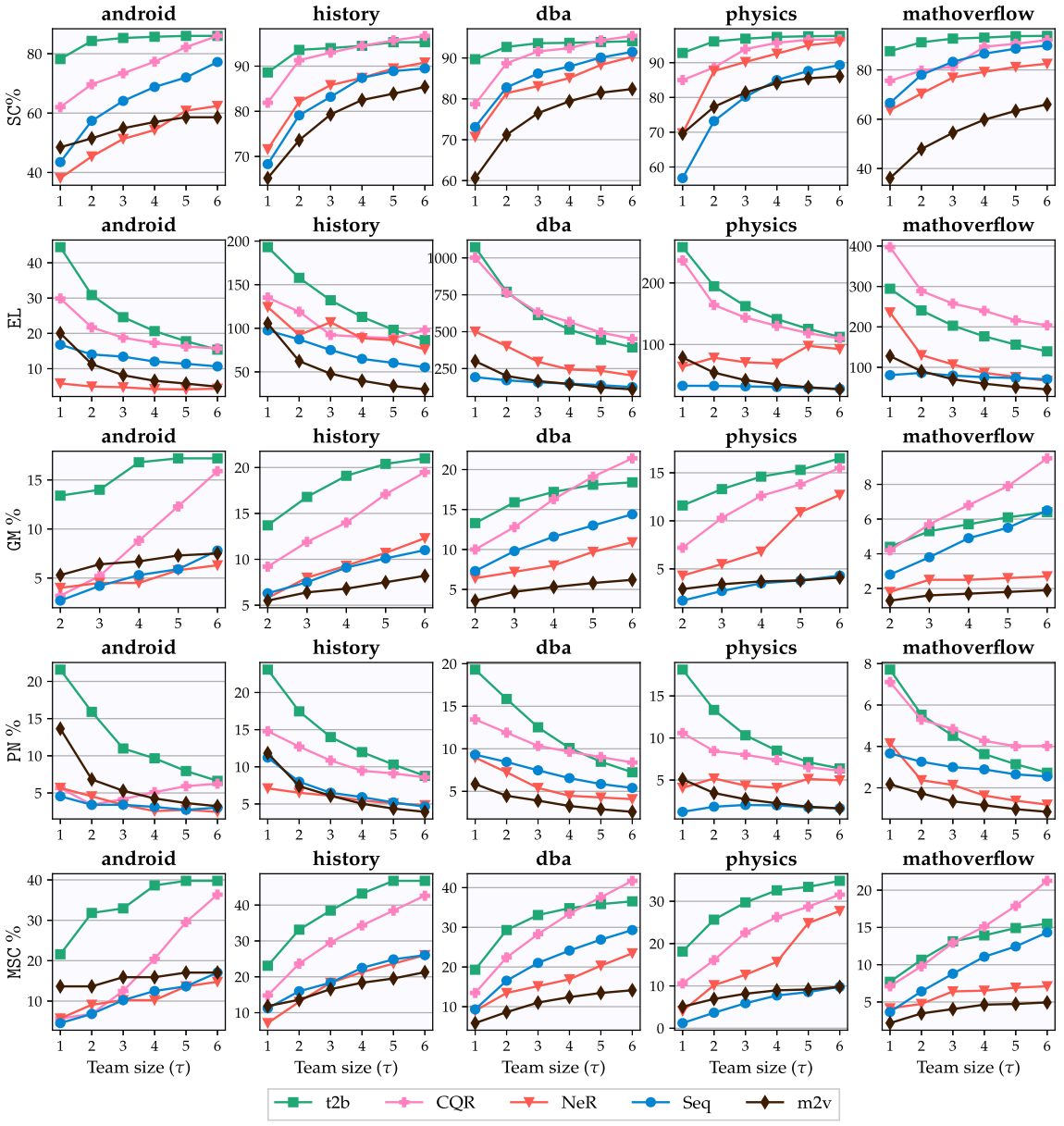

Fig. 2. The heterogeneous network graph for CQA. Panel (A) shows an example CQA network with two node types, i.e., questions and experts, along *question-expert (q—e)* relationships. The weight of each edge *q—e* indicates the difference between the up and down votes for expert *e* obtained by answering question *q*.

4. The proposed approach

Algorithm 1: Embedding-based Team Formation

Input: $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{W}), Q, q_{new}, \tau$.
Output: T
1 **begin**
2 $\quad \mathcal{G}_{embed} \leftarrow$ team2box($\mathcal{G}$).
3 $\quad L2R \leftarrow$ Learn2Rank($\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{W}$).
4 $\quad$ **foreach** $q \in Q$ **do**
5 $\quad\quad sim[q] \leftarrow L2R(q, q_{new})$.
6 $\quad$ **end**
7 $\quad topk_q \leftarrow \{$top k questions in Q with highest $sim\}$.
8 $\quad$ **foreach** $q' \in topk_q$ **do**
9 $\quad\quad embed_{q_{new}} += \left(sim[q']/\sum_{q \in topk_q} sim[q]\right) \times \mathcal{G}_{embed}[q']$.
10 $\quad$ **end**
11 $\quad topz_t \leftarrow$ top z teams with lowest $dist\left(C^{(team)}, embed_{q_{new}}\right)$.
12 $\quad$ T $\leftarrow$ top τ experts in $topz_t$.
13 $\quad$ **return** T.
14 **end**

We sketch the outline of our proposed approach based on Algorithm 1. First, our method maps the weighted heterogeneous network graph $\mathcal{G}$ into vector space, producing $\mathcal{G}_{embed}$ (Line 2), which is the core contribution of this paper and is further explained in details in Section 4.1. Our proposed method learns embeddings for teams, experts, and questions within the same embedding space. A team is considered as a sub-graph in CQA network constructed by a node of question type and its neighbor nodes.

Now, given new question q_{new}, we adopt a learning to rank (L2R) strategy to obtain the top k most similar questions to q_{new} (Lines 3–7) as explained in Section 4.2. The embedding vectors of the top k most similar questions from $\mathcal{G}_{embed}$ are used to map q_{new} into the same vector space (Lines 8–10). Finally, the distances between the embedding vectors of the teams in $\mathcal{G}_{embed}$ and the embedding vector of q_{new} are computed, based on which, the top τ experts are selected from the most similar teams to the question (Lines 11–12) in a way that each selected expert poses broader skills, and higher expertise related to the new question.

4.1. team2box: Embeddings for team formation

Our proposed team2box (Line 2 of Alg. 1) method preserves the structure of teams and the relationship between experts and questions in embedding space. We argue that it would not be sufficient to only capture the relationship between questions and experts to find a team. It is of paramount importance to also learn from past expert collaboration history when embedding experts and questions so as to preserve explicit team membership information when learning the embedding representations. As shown in Panel (B) of Fig. 2, the objective of team2box is to embed teams as regions in the embedding space and experts and questions as points (vectors) in the same space. Preserving teams' structures and their relationships has several advantages:

1. (1) it makes it possible to identify past relevant teams given an input question without having to compose teams from individual experts;
2. (2) it allows for determining appropriate team structure (e.g., ideal team size) based on how the input question relates to its closest teams in the embedding space;
3. (3) it ensures that team members exhibit complementary skill sets for answering a question, as opposed to having highly overlapping skills.

To build the embedding space, we first present how we design the loss function to embed the teams, and the loss function to embed experts and questions. Finally, we present the loss function for team2box for learning the full embedding space.

Team Embedding Loss Function. team2box models teams and their relations, i.e., intersection, as an undirected weighted graph, which we refer to as the *team network graph*. In this graph, the nodes denote teams and edges show non-empty intersection between the teams. The weight of each edge is defined as the fraction of common experts between the two endpoint teams. For example, the network graph of three teams in Fig. 2 Panel (A), i.e., $team_1 = \{e_1, e_2\}, team_2 = \{e_2, e_3, e_4\}$, and $team_3 = \{e_4, e_5\}$, is depicted in Fig. 2 Panel (D). There is no edge between $team_1$ and $team_3$ because their intersection is empty. The weight of $\frac{1}{4}$ for the $(team_1, team_2)$ edge is computed as $w_{1,2} = \frac{|team_1 \cap team_2|}{|team_1|+|team_2|-|team_1 \cap team_2|}$; where $|X|$ denotes the size of set X. Note that all teams in the network graph are unique. It means that there is no two teams with exactly the same members. Given the *team network graph*, our objective is to represent each team as a box using a *center* vector and an *offset*. A team i is specified in a d-dimensional embedding space as $i = (C^{(i)}, O^{(i)})$ where the center $C^{(i)}$ vector and offset $O^{(i)}$ define the team region as follows:

$$Box_i = \{v \in \mathbf{R}^d | dist(C^{(i)}, v) \leqslant O^{(i)}\}. \tag{1}$$

where $dist(C^{(i)}, v)$ denotes the distance between vector v and center vector $C^{(i)}$ and dist(.) is computed using Euclidean distance as:

$$dist(C^{(i)}, v) = \sqrt{\sum_{j=1}^{d}(C_j^{(i)} - v_j)}, \tag{2}$$

where $C_j^{(i)}$ and v_j is the j^{th} elements of vector $C^{(i)}$ and v.

In team2box, the offset $O^{(i)}$ is a constant proportional to the size while the center vector $C^{(i)}$ is learned using skip-gram with negative sampling over the *team network graph*. Given team i and j as positive training samples, and $k = 1, 2, \ldots, K$ as K negative samples, we minimize the loss as:

$$\begin{aligned} \ell_{box} = \quad & \Big(dist(C^{(i)}, C^{(j)}) - d_{ij}\Big)^2 + \\ & \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{C^{(k)}\sim P(C)}\Big[max\Big(0, d_{ik} - dist(C^{(i)}, C^{(k)})\Big)\Big]^2. \end{aligned} \tag{3}$$

where $dist(C^{(i)}, C^{(j)})$ is the distance between two centers and d_{ij} is a constant to control the overlap between the teams. For any two nodes in the *team network graph*, $d_{ij} = (1 - w_{ij})(O^{(i)} + O^{(j)})$ where w_{ij} is the weight of the edge for adjacent nodes and zero otherwise. Based on the weight function of the team network graph $w_{ii} = \frac{|team_i \cap team_i|}{|team_i| + |team_i| - |team_i \cap team_i|} = 1$. Thus, $d_{ii} = 0$ (exact match of the center vectors). Suppose three teams i, j, k with the same size, and $|team_i \cap team_j| \geqslant |team_i \cap team_k|$. It is captured by $w_{ij} \geqslant w_{ik}$ in the team network graph. During the model training it is translated to $d_{ij} < d_{ik}$ in the first part of Eq. 3. In other words, the overlapping between the embedding boxes of i and j should be higher than the boxes of i and k. Suppose team i and a negative sample team k, where $|team_i \cap team_k| = 0$. In such a case $w_{ik} = 0$, and $d_{ik} = O^{(i)} + O^{(k)}$. The second term in Eq. 3 penalizes the model when $dist(C^{(i)}, C^{(k)}) < d_{ik}$; otherwise there is no penalties due to no overlapping between the boxes of i and k. Note that Euclidean distance is used as $dist()$ function. The first term in the loss function tends to embed adjacent teams with common members as overlapping boxes in the embedding space, while the second term in Eq. 3 penalizes when disjoint teams are embedded as overlapping boxes. Furthermore, the percentage of overlapping section between the boxes of teams i and j is controlled by d_{ij} computed based on the percentage of their common experts, i.e. w_{ij}.

Experts and Questions Embedding Loss Function. Given graph $\mathcal{G}$, the d-dimensional latent representations are learned as $X \in \mathbf{R}^{N\times d}$ for all nodes of type question and expert such that it preserves their semantic and structural relationships. Note that here $N = n + m$ and $d \ll N$. To do so, parameters θ need to be learned to optimize:

$$\ell_{e,q} = \text{argmax}_\theta \sum_{v\in\mathcal{V}}\sum_{t\in T_v'}\sum_{v_t\in N_t(v)} log\ P(v_t|v;\theta). \tag{4}$$

where $T_v' = \{experts, questions\}$ and $N_t(v)$ is the set of neighbor nodes of type t of node v. Probability $P(v_t|v;\theta)$ is a softmax function defined as: $P(v_t|v;\theta) = \frac{exp(x_{v_t}.x_v)}{\sum_{\substack{u\in\mathcal{V}\\ \phi(u)\in T_t'}} exp(x_u.x_v)}$, where $\phi(u) : \mathcal{V} \to T_v$ mapping node type for each node u, and x_v is the embedding vector for node v. Computing softmax is demanding for large networks. Thus, we adopt a skip-gram with negative sampling strategy introduced in [44].

Overall team2box Loss Function. Given the loss functions in Eqs. 3 and 4, similar to [45], we combine the loss functions by linearly interpolating them: $\ell_{team2box} = \ell_{box} + \ell_{e,q}$. Once the embedding of teams, experts, and questions are learned, new teams can be formed for a new question based on the process explained earlier in Alg. 1. The model is optimized by using the ADAM stochastic gradient descent algorithm [46].

4.2. Question ranking

A learn to rank (*L2R*) model is trained using the textual information of existing questions and their answers. Given a question and its answers, the *L2R* model is trained to give higher ranks for the answers of the question compared to some random non-relevant answers. The trained model ranks existing questions given new question q_{new}. The embedding vectors of top k existing questions obtained by *team2box* are used to map the new question into the embedding space of teams, questions, and experts. Later, in the experiments, we will investigate how the effectiveness of the learn to rank method *L2R* can impact the performance of our method.

Table 2
Statistics of the datasets.

Dataset	Questions	Answers	Experts	Teams
android	882	2,136	1,018	857
history	1,697	4,702	1,164	1,575
dba	2,906	6,976	1,794	2,554
physics	4,912	12,559	2,938	4,613
mathoverflow	10,128	28,238	4,016	9,582

Table 3
Data prepossessing statistics.

Datasets	Answers		Question and Answer			
			Original		After Cleaning	
	All	Score >=4	Words	Chars	Words	Chars
android	3,728	2,136	6,862,586	39,675,435	205,154	1,405,186
history	6,839	4,702	5,447,670	32,987,040	916,589	6,778,405
dba	9,357	6,976	27,783,124	144,459,501	1,171,967	8,248,834
physics	19,475	12,559	31,327,261	197,412,782	2,142,405	15,566,073
mathoverflow	34,631	28,238	33,423,333	21,5342,932	3,887,787	28,315,688

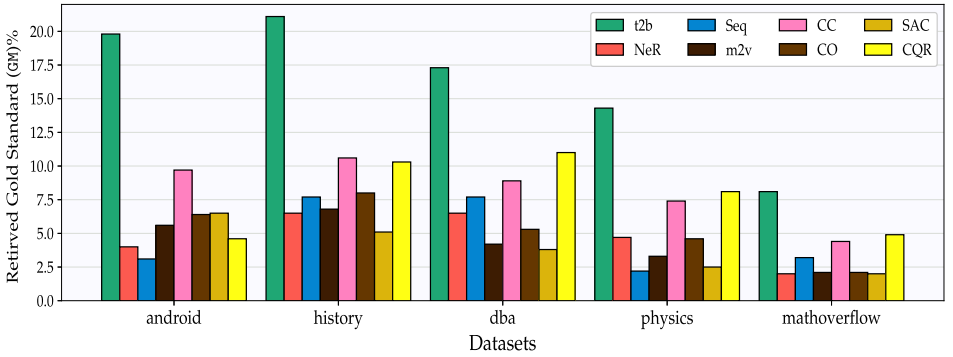

Fig. 3. Frequency of answer scores (upvote-downvote) for questions.

5. Experimental results

To investigate the performance of team2box, we carry out extensive experiments on real-world datasets from a variety of domains with different sizes, and report the results along with our observations. The pre-processed datasets along with our code and the output data are available on-line[2].

Datasets. We perform experiments on several real-world datasets from Stack Exchange, released in Sept 2019, whose properties are summarized in Table 2. In preprocessing, all stop words and special characters are removed and only questions with a minimum of two answers are retained. Also, answers with voting scores less than four are removed. Table 3 summarizes the properties of the datasets before and after performing preprocessing. The voting scores (upvotes-downvotes) of answers and their frequencies are shown in Fig. 3. The statistics show that more than 80% of answers in each dataset have voting scores in the range of $[4, 20]$.

Experimental Setup and Evaluation Metrics. In our experiments, we use K-NRM [47] with the default setting as the learning to rank method to retrieve relevant questions for a new question (Lines 2–5 of Algorithm 1). The first 100 words from the title and body of each question, and the content of each answer are used to create a pair of query and document as a training sample to feed K-NRM with a randomly initialized word embedding. Moreover, in all the methods, the embedding dimension of nodes is set to 128. The embedding vectors of nodes are randomly initialized. The other parameters of the baseline methods remain as proposed by the authors. In Algorithm 1, we set $k = 10, z = 11$ for all the datasets. Recall that z denotes the number of discovered top teams used to choose the members of new teams. In our experiments, we set the maximum number of team size as six due to the fact that the average team size is 5.28 in our datasets. Furthermore, based on the characteristics of our datasets, we empirically found that z needs to be set at 11 in order to guarantee that there will be at least six experts in the retrieved top z teams. The center vectors for teams are randomly initialized and the offset of team i is

[2] https://github.com/team-formation/team2box

defined as its size divided by twenty. In Eq.s 3, the number of negative samples, i.e. K, is in the range 2–10. Furthermore, the noise distribution, i.e. $P(X)$, is considered as the unigram distribution $U(X)$ raised to the 3/4rd power ($X = C, u$).

Given the Stack Exchange datasets, our gold standard teams are defined as the set of users who have answered a given question. Our goal is to predict the team for each new incoming question in the test dataset. For each dataset, 90% of the questions are used for training and the remaining 10% are utilized for testing the performance of the methods.

Inspired by widely adopted metrics used in team formation literature [9,40,8,37,7,8], we utilize the following metrics in our evaluations:

(1) Skill Coverage (SC): we measure the number of common tags between the new question and existing questions answered by the members of the retrieved team. Ideally, a team should have full coverage, showing that the team members should have covered all the required tags by the new question in the questions they answered in the past. Let n be the number of test questions. Given $tg^{(i)}$ as a set of tags for question q_i, *skill coverage*, denoted as SC, is computed as:

$$\mathtt{SC} = \frac{1}{n}\left(\sum_{k=1}^{n}\frac{1}{|tg^{(k)}|}|tg^{(k)} \cap \{\bigcup_{q_i \in Q^{\mathrm{T}_k}} tg^{(i)}\}|\right) \times 100, \tag{5}$$

where T_k is the discovered team for question k and Q^{T_k} is the set of past questions answered by experts in T_k, and $|Y|$ is the size of set Y. The range of SC values is in $[0, 100]$ where larger values are more desirable.

(2) Expertise Level (EL): Akin to the citation counts in publications, the score of each answer shows its quality and popularity among the community members. We compute expertise level as follows:

$$\mathtt{EL}(q_i, \mathrm{T}_i) = \frac{1}{\beta|\mathrm{T}_i|}\sum_{e \in \mathrm{T}_i}\sum_{t \in tg^{(i)}}\sum_{\substack{q_j \in Q^{(e)},\\ t \in tg^{(j)}}} s_j, \tag{6}$$

where $Q^{(e)}$ is a set of past questions answered by expert e, and s_j is the score of the answer published by e for question q_j. Given $tg^{(i)}$ as a set of tags for question q_i, and β as the number of tags of the new question, i.e., $\beta = |tg^{(new)}|$, the EL (q_i, T_i) metric measures the average score per tag in question q_i obtained by the experts in team T_i by answering past questions with the same tags in q_i. We compute the average of EL over n test questions as: $\frac{1}{n}\sum_{k=1}^{n}\mathtt{EL}(q_k, \mathrm{T}_k)$.

(3) Gold Standard Team Match (GM): We further compute the match between discovered teams and the actual answerers of test questions as gold standard teams. Suppose T and T_g denote the discovered and gold standard teams for a given question, respectively. Metric gold standard team match denoted by GM is computed as $\mathtt{GM} = |\mathrm{T} \cap \mathrm{T}_g|/|\mathrm{T}_g|$. Metric GM shows how likely the discovered experts worked as a team in reality to answer a given question. The GM metric is equivalent to the recall measure in the IR community.

(4) Precision at N (PN): This metric measures the percentage of discovered experts, i.e. T, which match with the actual answerers of each test question, i.e. T_g, and is computed for n test questions as follows [5]:

$$\mathtt{PN} = \frac{1}{n}\sum_{k=1}^{n}\frac{|\mathrm{T}_k \cap \mathrm{T}_{g_k}|}{|\mathrm{T}_k|}, \tag{7}$$

where T_k is the recommended answerers of test question k, and T_{g_k} is its actual answerers.

(5) Matching Set Count (MSC): This metric reveals the percentage of test questions in which at least one of their recommended answerers matches with their actual answerers. MSC is computed for n test questions as follows:

$$\mathtt{MSC} = \frac{1}{n}\sum_{k=1}^{n} f(\mathrm{T}_k, \mathrm{T}_{g_k}), \tag{8}$$

and

$$f(\mathrm{T}_k, \mathrm{T}_{g_k}) = \begin{cases} 1, & \text{if} \mathrm{T}_k \cap \mathrm{T}_{g_k} \neq \phi \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

(6) Workload (WL) measures how experts are assigned to questions in terms of their workload. Ideally, none of the experts should be overloaded with tasks or being unfairly under-employed. Given n new test questions, the metric WL is computed as:

$$\mathtt{WL} = \frac{\sum_{i=1}^{n}|\mathrm{T}_i|}{n * n_e}, \tag{10}$$

where T_i is the team assigned to test question q_i, and n_e is the total number of unique experts utilized in the n teams. The WL metric computes the average number of teams in which each retrieved expert collaborates across the n test questions. The values of WL can range between $[\frac{1}{n}, 1]$ where the more desirable value, i.e., $\frac{1}{n}$, indicates that each expert is only assigned for one

Table 4
Example of computing evaluation metrics on a team formed by `team2box` for a test question of android dataset.

Test question q_{new}**Title**: How can I fix the GPS on my Samsung Galaxy S?
Body: As has been well documented, the Galaxy S phones have terrible GPS functionality.
It works for a minority, but it is slow/inaccurate for some and for others
it just doesn't work at all. How can I fix this?
Tags set $tg^{(new)}$: {gps, samsung-galaxy-s}
Actual answerers T_g={Matthew Read-Id:1465, Flow-Id:440, ce4-Id:15713}

Retrieved team T={Matthew Read-Id:1465, GAThrawn-Id:156, Lie Ryan-Id:482} **1465's overlapping tags/scores**: {(gps, scores:45), (samsung-galaxy-s, scores:62)}
156's overlapping tags/ scores: {(samsung-galaxy-s, scores:45)}
482's overlapping tags/ scores: {(gps, scores:32), (samsung-galaxy-s, scores:4)}
$\{\bigcup_{q_i \in Q^T} tg^{(i)}\}$={samsung-galaxy-s, gps}. Recall that Q^T is a set of questions answered by members in T.
$\mathtt{SC} = (|tg^{(new)} \cap \{\text{samsung} - \text{galaxy} - \text{s}, \text{gps}\}|/|tg^{(new)}|) \times 100 = \frac{2}{2} \times 100 = 100\%$.
$\mathtt{EL}(q_{new}, T) = \frac{1}{|T|}\sum_{e \in T}\sum_{t \in tg^{(new)}}\sum_{\substack{q_j \in Q^{(e)}, \\ t \in tg^{(j)}}} s_j = \frac{1}{2*3}[(45+62)+45+(32+4)] = 31.33$.
$\mathtt{GM}=\frac{|T \cap T_g|}{|T_g|} = \frac{1}{3}$, $\mathtt{PN}=\frac{|T \cap T_g|}{|T|} = \frac{1}{3}$, $\mathtt{MSC}=\frac{1}{1} \times 1 = 1$, $\mathtt{WL}=\frac{3}{1\times 3} = 1$.

team among n discovered teams. In contrast, the upper bound value, i.e., 1, shows that all discovered teams have the same members, which is the worst case scenario in terms of workload for the selected team members.

Table 4 elaborates on the computation of the evaluation metrics using a test question from the android dataset. For the sake of illustration, `team2box` is used to predict a potential team with three members for this question. The team in the figure achieves 100% skill coverage (`SC`), which means that the set of tags of existing questions answered by the members of the retrieved team has complete overlap with the set of tags of the new test question. The value of $\mathtt{EL} = 31.33$ shows that each team member obtained on average 31.33 votes by answering existing questions with tag(s) in the tag set of the test question. $\mathtt{GM} = 1/3$ indicates that one third of members of T_g, are among the recommended answerers of the question. $\mathtt{PN} = 1/3$ shows that one third of the recommended answerers are from the actual answerers of the question. Metric $\mathtt{MSC} = 1$ reveals that in all of the test questions at least one of their recommended experts are from their actual answerers. Finally, $\mathtt{WL} = 1$ indicates that each retrieved expert collaborates in all of the test questions; in other words, the same experts are assigned to all teams.

We would like to highlight that Table 5 summarizes how the evaluation metrics used in our experiments evaluate the main three objectives of team formation task in a CQA system that were introduced in the Problem definition in Section 3.

Baselines. We adopt two sets of baselines for comparing our work in the experiments:

Expert finding: We compare the proposed `team2box` with the state-of-the-art *expert finding* methods, i.e., `NeRank` [1] and `Seq` [2]. Both `NeRank` and `Seq` methods utilize the structural information of a heterogeneous network built using the relationships between the existing questions, their askers, and answerers, and learn the representation of nodes of this network in embedding space. Then, such structural latent representations along with textual information of existing questions are used to predict the best answerer for the new question. `NeRank` utilizes the title and body of the questions as textual information. In contrast, `Seq` uses question tags as textual information.

Algorithm 1 can be replicated by replacing our proposed `team2box` method (Line 2) with any heterogeneous graph embedding method. We additionally utilize `metapath2vec` [43] as a state-of-the-art heterogeneous graph embedding method to replace Line 2 of Algorithm 1 and hence produce a third baseline.

We also compare our method with the collaborative question routing method, called here `CQR` [5]. Method `CQR` is a greedy approach and employs several heuristics to extract expertise of experts and their availability and compatibility with each other from a CQA environment. Given a new question, to build a collaborative group of experts of size n, it first adds an expert with higher expertise and availability into the team. Then, it computes the compatibility of each unselected expert with the members of the team. A new member is chosen with higher expertise and compatibility with the team. This process

Table 5
The coverage of team formation objectives by the evaluation metrics.

Objectives	Metrics				Description
	SC	EL	GM	PN	
O1	✓				O1 is quantified by the common tags of past questions answered by members of discovered team T and q_{new}.
O2		✓			O2 is quantified by the voting scores of past answers provided by members of discovered team T.
O3			✓	✓	O3 is quantified by the common past questions answered by pairs of experts in T.

is repeated $n-1$ times to have a team of size n. In our implementation, it is assumed that all the experts are available to answer new questions. One could view this as a limitation of our work that needs to be addressed in future work. One simple approach can be to filter inactive experts from the system and form teams from available and active experts.

Classic team formation: Due to the resemblance of our work to the classic team formation problem, we also compare our work with three state-of-the-art team formation techniques, namely `CC` [6], `CO` [9], `SA-CA-CC` [10]. Given a new question and its tags, these methods find a set of experts that collectively covers the tags. Furthermore, `CC` finds a team while maximizing the collaboration level among the members. `CO` finds a team while maximizing collaboration level among team members and their expertise level. `SA-CA-CC` finds a team while maximizing the collaboration level and the expertise level of team members and their connectors in the network. Note that, all these methods are heuristics-driven and search locally on the graph to find a team. Therefore, the output team is often not optimal given the local search on the graph.

Random methods: In addition, two naive approaches are considered in our comparison. The first random method, referred to as `rnd`, forms a team for a given new test question by selecting experts at random without replacement from all experts in the dataset. The second method improves `rnd` by choosing experts from the set of experts with at least one common tag with the question, referred to as `rct`.

5.1. Comparison with expert finding baselines

This section covers the comparison of `team2box` and the state-of-the-art *expert finding* techniques, i.e. `NeRank` [1] and `Seq` [2], `metapath2vec` [43], and `CQR` [5]. The performance of these methods are presented in Fig. 4. In each dataset, teams with six different sizes are considered, i.e., $\tau = 1, 2, .., 6$. Top $k(= 1, 2, .., 6)$ experts retrieved by the *expert finding* methods are used to form a team with a size of k.

The experiments indicate that `team2box` has consistently provided superior performance in terms of `SC`, `EL`, `GM`, `PN`, and `MSC` evaluation metrics on the majority of datasets. Furthermore, as expected the random methods show inferior results. The reasons for these observations are summarized as follows:

(A) `team2box` achieves higher skill coverage, i.e. `SC`, compared to the other methods primarily because it uses not only the content of existing questions, but also the content of the experts' answers to the questions. In contrast, `NeRank` [1] only uses the latent representation of the content (tags, title, and body) of existing questions to relate them to the content of the new question. Similarly, `Seq` [2] utilizes only the tags of past questions and learns their latent representations to retrieve relevant experts to the new question. Both methods suffer from ignoring the content of the experts' answers to existing questions. Given our approach considers both questions' content and the experts' answers to these questions, it is expected that it would be able to learn meaningful relations between experts, their past answers and the questions; hence, facilitating a more efficient retrieval of relevant experts for a new question. This translates into a higher skill coverage for `team2box` compared to those other methods. This is empirically supported by the experiments in which `team2box` obtains on average 38.7%, 24.3%, 19.4%, and 6.24% improvement on skill coverage on all the datasets compared to `metapath2vec`, `NeRank`, `Seq`, and `CQR`, respectively. The random method, `rnd`, obtains the worst performance on all data. As expected, the `rct` method improves `rnd`. However, both random methods exhibit inferior performance compared to the other methods.

(B) In terms of expertise level of the retrieved teams, i.e. `EL`, `team2box` outperforms the other methods. Akin to `NeRank` and `Seq`, `team2box` models the existing questions and experts (answerers) and their relationships as a heterogeneous network. However, `team2box` adds the voting scores of answers as weights of links in the network, which has been overlooked by both `NeRank` and `Seq`. The voting scores of experts' answers can be translated as their expertise level. Thus, `team2box` learns latent representations of the existing questions and experts using the voting scores as weights of links in CQA networks compared to the other two baselines that utilize unweighted networks. Note that `CQR` utilizes the voting scores of the past answers of experts to estimate the expertise score of experts given a topic. It uses such scores to chose members of new teams. Based on the experiments, `team2box` improves the expertise level of the retrieved teams, on average, by 216%, 154%, 209%, 5.7% compared to `metapath2vec`, `NeRank`, `Seq`, and `CQR` respectively, on all datasets. As shown in the plots, the expertise level of experts retrieved at random by `rnd` and `rct` is close to the lowest `EL` value (zero) for all the datasets.

(C) We further investigate the match between the members of discovered teams and the gold standard teams' members. For each test question, all of its answerers are considered as a gold standard team. Then, the methods retrieve members of teams among all experts (answerers) in each dataset. Then, the ratio of the intersection between the discovered team with a specific size ($\tau = 1, .., 6$) and the gold standard one is computed. The average of such intersection ratios over the test questions is reported as gold standard match, i.e. `GM`, for each method in the forth row of Fig. 4. We make two main observations as follows:

- i) `team2box` retrieves the actual answerers of test questions more effectively compared to the existing methods. Based on the figure and on average 7.46% of the first selected members of discovered teams are among the actual answerers of the test questions. In other words, `team2box` retrieves the true answerers of the questions by around 2.49 times better than the other methods;

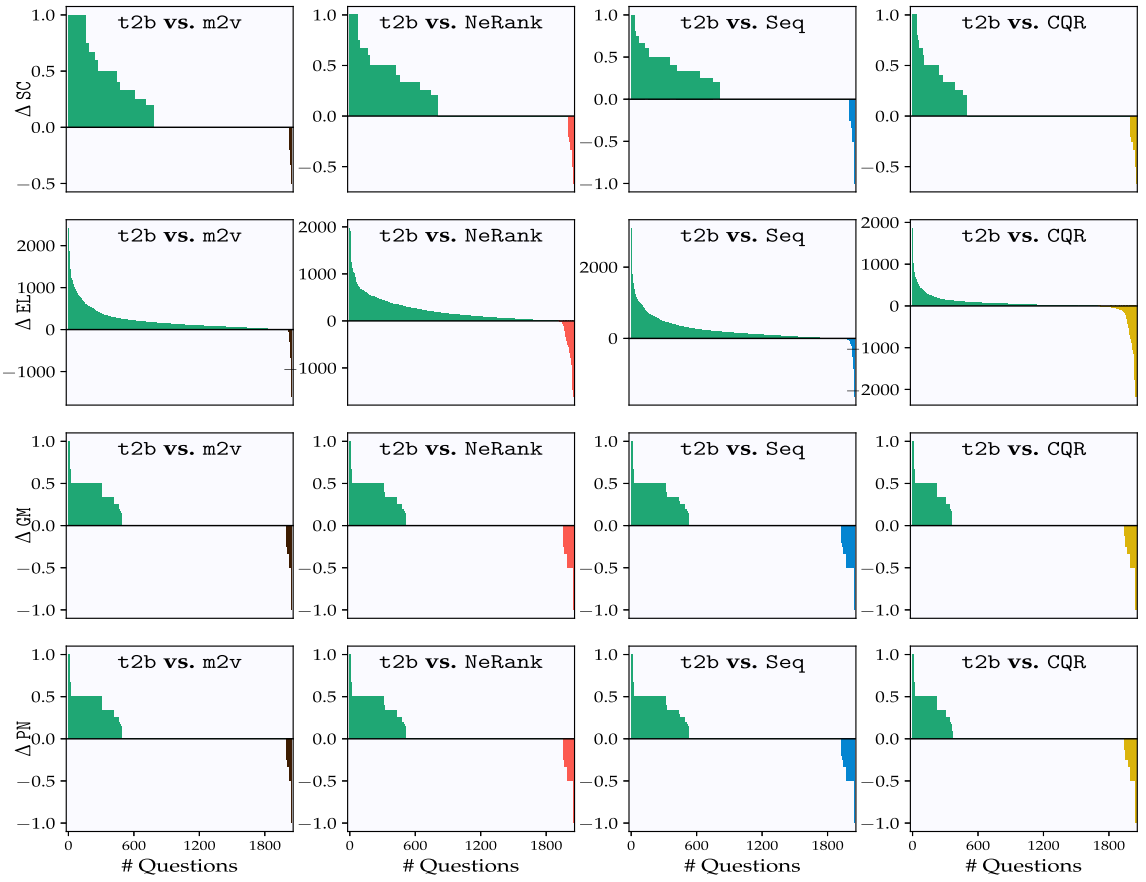

Fig. 4. Comparison of `team2box` (`t2b`) with the state-of-the-art *expert finding* methods, i.e. `NeRank` [1], `Seq` [2], heterogeneous graph embedding technique, i.e. `metapath2vec` denoted [43], existing collaborative expert finding technique [5], and random methods `rnd` and `rct`., based on skill coverage (`SC`), expertise level (`EL`), precision at N (`PN`), and matching set count (`MSC`). `GM` shows the percentage of the actual answerers of test questions retrieved by the methods. Similarly, `PN` reveals the percentage of recommended experts which are among the actual answerers. `MSC` shows the fraction of test questions in which at least one of their recommended answerers is among their actual answerers.

- ii) as expected, the number of actual answerers discovered by `team2box` constantly grows by almost the same ratio over all datasets while increasing the size of teams from one to six. As shown in the figure, on average 15.90% of actual answerers of questions are among the members of teams with size six discovered by `team2box` which is doubled compared to the case when the teams size is one. Such results are 5.58%, 8.98%, 8.79%, and 10.36% for `metapath2vec`, `NeRank`, `Seq`, and `CQR`, respectively.

We note that while the random `rct` method does not show competitive performance on the `GM` metric on four out of the five datasets, it shows good performance on the android dataset. We believe this is due to the size of the android dataset, which is the smallest dataset in our experiments. When the size of the dataset is small (hence a smaller number of possible experts to choose from), a method that randomly chooses an expert with the same set of tags as the question, as done in `rct`, has a higher likelihood of matching the gold standard user in the dataset. However, as the size of the datasets becomes larger, as is the case in the other four datasets, `rct` shows inferior performance on `GM`.

(D) We observe similar behaviour among the methods using `PN` compared to `GM`. Our `team2box` method outperforms the baselines in terms of `PN`. Recall the that `PN` measures what percentage of recommended experts of new teams are among the actual answerers of test questions. The experiments indicate that `team2box` discovers teams with on average 43.69% higher `PN` compared to our best baseline, i.e. `CQR` on all datasets. The improvements are 204.2%, 191.4%, and 146.9% compared to the `Seq`,`m2v`, and `NeRank` methods, respectively on all datasets.

(E) The proposed `team2box` method obtained superior results based on `MSC` compared to the baselines. The experiments show that on average in 29.17% of the discovered teams, with sizes ranging from one to six, at least one of their members is among the actual answerers of test questions. It is 198.6%, 184.7%, 140.4%, and 46.18% improvements compared to `Seq`, `m2v`, `NeRank`, and `CQR`, respectively.

We also compare the methods based on the workload metric (`WL`) and summarize the results in Table 6. The results show that the workload for experts retrieved by the proposed method is always less than 0.07 for all of the datasets, which is lower than `CQR` with 0.25, `NeR` with 0.33 and `m2v` with 1.0. As expected the random methods, `rnd` and `rct`, form teams with near

minimum WL values, i.e. each expert participates at answering one question; however, this is due to the random assignment nature of these methods and does not speak to the quality of the generated teams as observed on the other evaluation metrics.

5.2. Comparison with team formation baselines

We further compare the performance of team2box to three related team formation techniques called CC [6], CO [9], SA-CA-CC [10]. Given team size τ is not an input to team formation techniques and they retrieve teams with different sizes for different test questions, we report the results of each method in a separate table. The parameters of the proposed method is tuned to obtain teams with the same size as those built using the baselines. We report the results in Tables 7–9. Note that the results for team2box in these tables may be different from those in Fig. 4 for the same datasets due to using different team sizes. We summarize our observations as follows:

(A) team2box achieves a higher performance compared to the baselines on the different datasets for all of the metrics. In classic team formation, the size of the team is not an input parameter and the algorithms add members to the team till all the required skills are covered by the team members. However, since such methods restrict the search space to a local sub-graphs, they do not show high performance on skill coverage. In contrast, team2box learns the embeddings based on the

Table 6
Comparison of the methods based on the workload WL.

Dataset	Methods						
	rnd	rct	m2v	Seq	NeR	CQR	t2b
android	0.012	0.013	1.0	0.015	0.11	0.13	0.06
history	0.006	0.007	0.048	0.023	0.17	0.14	0.07
dba	0.004	0.004	0.04	0.008	0.33	0.25	0.06
physics	0.002	0.003	0.02	0.004	0.2	0.09	0.05
mathoverflow	0.001	0.001	0.005	0.002	0.08	0.08	0.02

Table 7
team2box vs. CC [6].

Datasets	SC(%)		EL		GM(%)		PN(%)		MSC(%)	
	CC	t2b	CC	t2b	CC	t2b	CC	t2b	CC	t2b
android	74.3	**85.7**	13.6	**20.6**	14.8	**16.8**	**10.04**	9.66	34.94	**38.64**
history	89.4	**94.5**	81.6	**113.0**	14.4	**19.1**	**14.01**	11.98	37.28	**43.2**
dba	86.4	**93.6**	466.6	**612.2**	13.0	**15.9**	12.45	**12.53**	30.5	**33.1**
physics	92.5	**96.9**	110.9	**162.3**	9.5	**13.3**	9.5	**10.32**	22.86	**29.74**
mathoverflow	86.4	**92.7**	212.1	**203.0**	**5.5**	5.3	**6.46**	4.51	**14.56**	13.14

Table 8
team2box vs. CO [9].

Datasets	SC(%)		EL		GM(%)		PN(%)		MSC(%)	
	CO	t2b	CO	t2b	CO	t2b	CO	t2b	CO	t2b
android	81.4	**86.0**	11.4	**15.4**	16.8	**17.2**	**8.41**	6.63	38.55	**39.77**
history	90.2	**95.3**	64.8	**98.2**	15.4	**20.4**	9.71	**10.3**	37.28	**46.75**
dba	90.2	**94.1**	264.4	**393.5**	13.9	**18.4**	6.35	**7.24**	31.91	**36.55**
physics	94.1	**97.8**	63.4	**101.3**	11.0	**16.7**	4.69	**5.56**	25.51	**35.23**
mathoverflow	87.5	**94.0**	89.2	**125.7**	4.8	**6.6**	2.09	**2.41**	12.56	**15.91**

Table 9
team2box vs. SA-CA-CC (SAC) [10].

Datasets	SC(%)		EL		GM(%)		PN(%)		MSC(%)	
	SAC	t2b	SAC	t2b	SAC	t2b	SAC	t2b	SAC	t2b
android	80.2	**86.0**	7.6	**11.9**	**19.1**	17.2	**6.61**	4.97	**43.37**	39.77
history	95.2	**95.3**	57.3	**69.1**	19.8	**21.3**	**7.32**	6.82	45.56	**47.34**
dba	88.7	**94.1**	251.8	**393.5**	13.5	**18.4**	5.76	**7.24**	32.62	**36.55**
physics	91.3	**97.8**	24.3	**78.4**	9.7	**17.1**	2.61	**4.22**	22.24	**36.25**
mathoverflow	88.9	**94.1**	70.0	**114.3**	5.4	**6.7**	2.0	**2.12**	14.16	**15.91**

whole network and thus discovers teams with high skill coverage. The results indicates that team2box builds teams with, on average, a 5% higher skill coverage compared to the team formation methods.

(B) Our team2box employs more knowledgeable experts in the field of new posted questions compared to the team formation baselines. As reported in the tables, the expertise level of teams formed by team2box is on average 32.64%, 47.23%, and 83.85% higher than CC, CO, and SA-CA-CC respectively on all datasets.

(C) team2box retrieves more actual answerers of the questions as members of discovered teams compared to the baselines. The experiments indicate that team2box employs the actual answerers of the questions around on average 2.91% more than the baselines to form new teams.

(D) Considering PN, our method is the second best method among the team formation baselines. Our proposed method builds teams with on average 6.55% and 12.34% superior and 7.83% worse results in terms PN compared to baselines CO, SAC, and CC, respectively.

(E) Our proposed team2box outperforms the baselines in terms of MSC. The experiments reveal that team2box achieves on average 21.57%, 16.6%, and 11.06% superior results compared to CO, SAC, and CC on all datasets, respectively.

The experiments indicate that compared to existing baseline techniques in CQA platforms, team2box retrieves a group of experts that can collectively cover the skills required to answer a new question while enjoying a high likelihood of positive collaboration between the team members.

5.3. Comparison with Gold Standard Teams

To further evaluation, the methods are compared when they retrieve teams with the same size as the gold standard ones. Given a test question, the methods discover teams with the same size as the number of actual answerers of the test question. Then, the average of the percentage of overlapping between the discovered teams and gold standard ones, i.e. GM and PN metrics, is computed for each method. Note that GM = PN when the size of a recommended team for the questions are the same as the actual teams. The results are reported in Fig. 5.

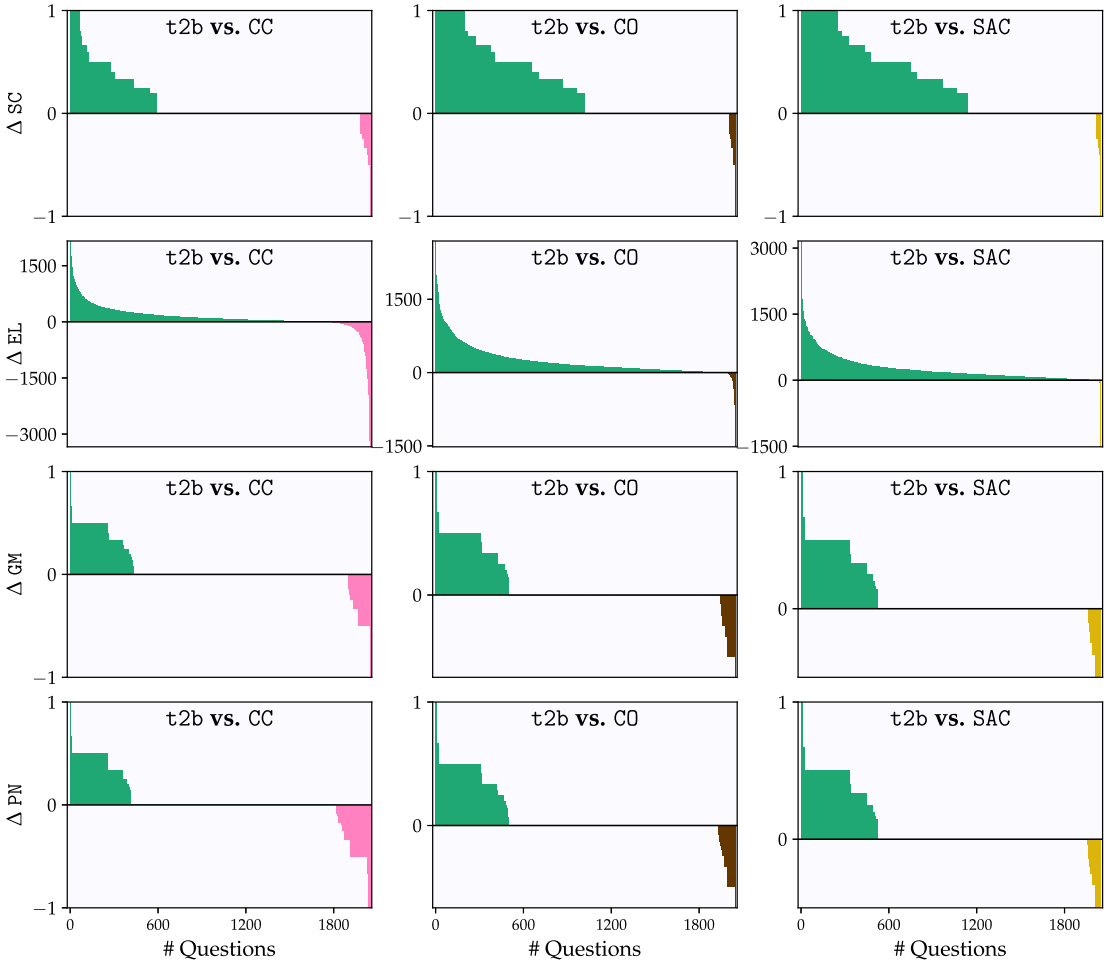

Fig. 5. Comparison of the retrieved teams with gold standard teams.

The experiments reveal that team2box outperforms the other existing methods, and on average 16.1% of its discovered teams' members are among the members of gold standard teams on all datasets. Compared to the second most efficient method, i.e. CC with 8.2%, team2box achieves around 96.34% improvements in terms of matching discovered teams with gold standard teams, i.e. GM.

We also report the performance of our team2box compared to the baselines methods in terms of different team formation metrics over all test questions in Figs. 6 and 7. In the plots, a positive value indicates that team2box outperforms existing methods. Similarly, negative values, i.e. $\Delta < 0$, show existing methods achieve better results compared to team2box. When $\Delta = 0$, the methods obtain the same results.

team2box **vs. expert finding baselines**: Fig. 6 compares the results of team2box with the expert finding baselines. The panels in the first row of the figure compare the methods in terms of skill coverage metric denoted by SC. The experiments indicate that team2box obtains better results on average 35.32% and worse results on only 2.04% of test questions compared to the baselines. The reason for such an observation is that team2box uses the latent representations of the content of the answers and the questions to connect new question into existing one. Thus, members of the teams close to the new question in the embedding space have a higher chance to possess the skills required by the new question.

We also compare the methods in terms of the expertise level, i.e. EL, of members of retrieved teams in the second row panels. The experiments show that team2box achieves superior results on 97.31%, 93.12%, 92.78%, and 74% of the test questions compared to m2v, Seq, NeRank, and CQR, respectively.

To understand the performance of the methods to discovered the true answerers of test question among all experts in each dataset, the difference between metric GM of the teams discovered by team2box and the other methods is computed for all questions and reported in the third row of the figure. The experiments indicate that team2box achieves superior

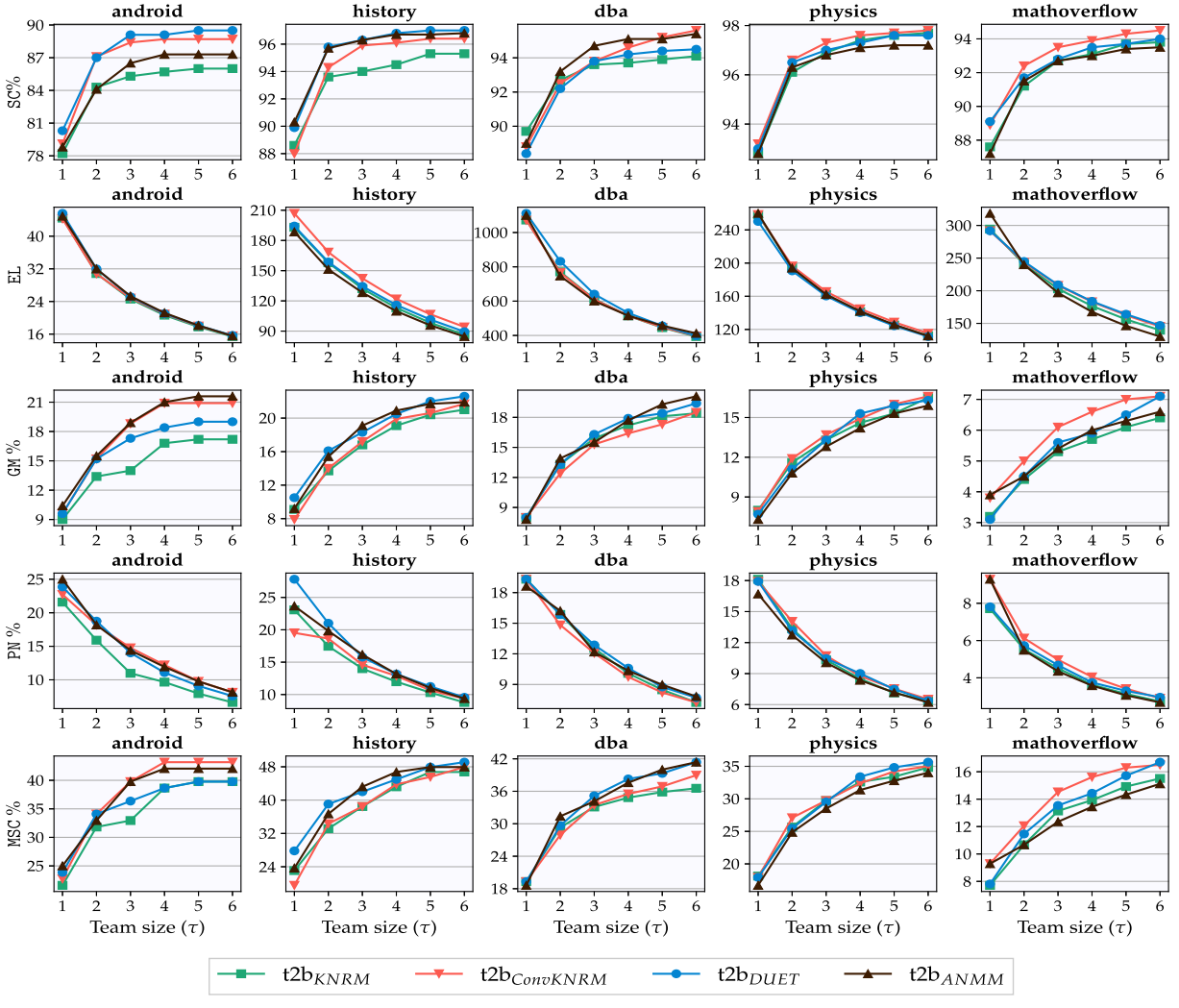

Fig. 6. Improvement ratios of proposed t2b vs. the expert finding baselines based on the team formation metrics over all test questions of the datasets.

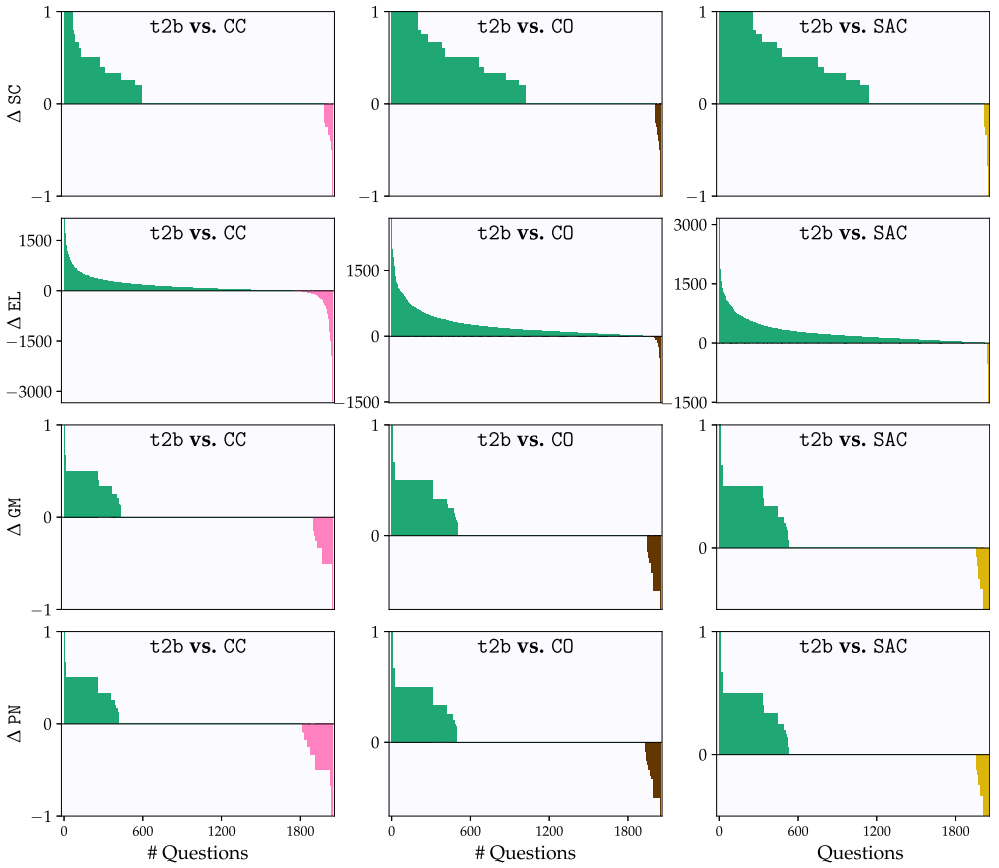

Fig. 7. Proposed team2box vs. the classic team formation baselines based on the improvements of the team formation metrics over all test questions of the datasets.

results in terms of GM on average in 23.10% of all test questions, and obtains the same GMs as the baselines on 72.32% of the questions. It means that the team discovered by team2box has a higher chance of occurring in reality compared to the ones formed by the baselines.

The experiments also reveal that our method outperforms the best baseline, i.e. CQR, by achieving superior results in 17.95% and equal results in 76.92% of test question in terms of the PN evaluation metric.

team2box **vs. team formation baselines**: The comparison of team2box against the team formation baselines in terms of the evaluation metrics is illustrated in Fig. 7. The methods are tuned to discover teams with the same size as the number of actual answerers of test questions. Then, the evaluation metrics are computed based on the teams discovered by each method. We make several observations as follows.

- Teams retrieved by team2box possess a higher chance of collectively having skills required to answer new questions compared to the baselines. As shown in the first row panels of Fig. 7, our team2box builds teams with higher skill coverage, i.e. SC, on average in 44.66%, and the same SC as the baselines in 53% of the test questions.
- Given a new question, the expertise level of experts retrieved by team2box is higher than the ones discovered by the team formation techniques in the fields required by the question. The experiments indicate that team2box outperforms the baselines in terms of EL on average in 93.43% of the test questions.
- Experts retrieved by team2box have high willingness to cooperate with each other as a team.
- team2box discovers more realistic teams compared to the baselines. The experiments show that 23.7% of the teams formed by team2box contain a larger number of actual answerers of the test questions than the teams built by the baselines which is reported in the last row of the figure.
- Considering PN, our proposed method outperforms the baselines in 23.38% of the test questions by discovering new teams with more members from their actual answerers. The experiments indicate that the baseline methods obtain similar results in 69.41% of the test questions.

5.4. Impact of learn to rank methods

Here we investigate the impact of using different learn to rank techniques on the performance of proposed method. Recall that our technique utilizes a learn to rank method to find top-k similar existing questions to the new question (Lines 3–5 of Algorithm 1). In our experiments so far we have only reported our results based on K-NRM. Now, in order to study the impact of alternative learn to rank methods, we employ other techniques such as DUET [48],ANMM [49], and ConvKNRM [50]. Note that ConvKNRM is a convolutional variant of method K-NRM. We used the same experimental settings for our proposed method for all three learn to rank techniques. The best results over five independent repetitions of the models are reported in Fig. 8. The literature has reported that ConvKNRM [50] obtains superior rank of documents compared K-NRM. As shown in the figure, the performance of our proposed method increases by better ranking existing questions given new questions by employing ConvKNRM. The experiments show that employing ConvKNRM and DUET increases the performance of our method by roughly 1%, 2%, 7%, and 5% in terms of evaluation metrics SC, EL,GM, PN, and MSC on all datasets, respectively. Furthermore, the performance of using ANMM is almost the same as K-NRM. As such, we conclude that the effectiveness of the learn to rank method does impact the performance of our proposed method. However, given our results reported in this paper are based on K-NRM, the results of our method would only improve if more recent learn to rank methods are employed.

5.5. Discussions

We have employed five team formation metrics to investigate the performance of our proposed approach on five real-life datasets from a variety of domains. Our experiments show that team2box is able to address the limitations introduced in the introduction section, namely $\ell 1 - 3$. To overcome $\ell 1$, team2box discovers a group of experts instead of finding an individual

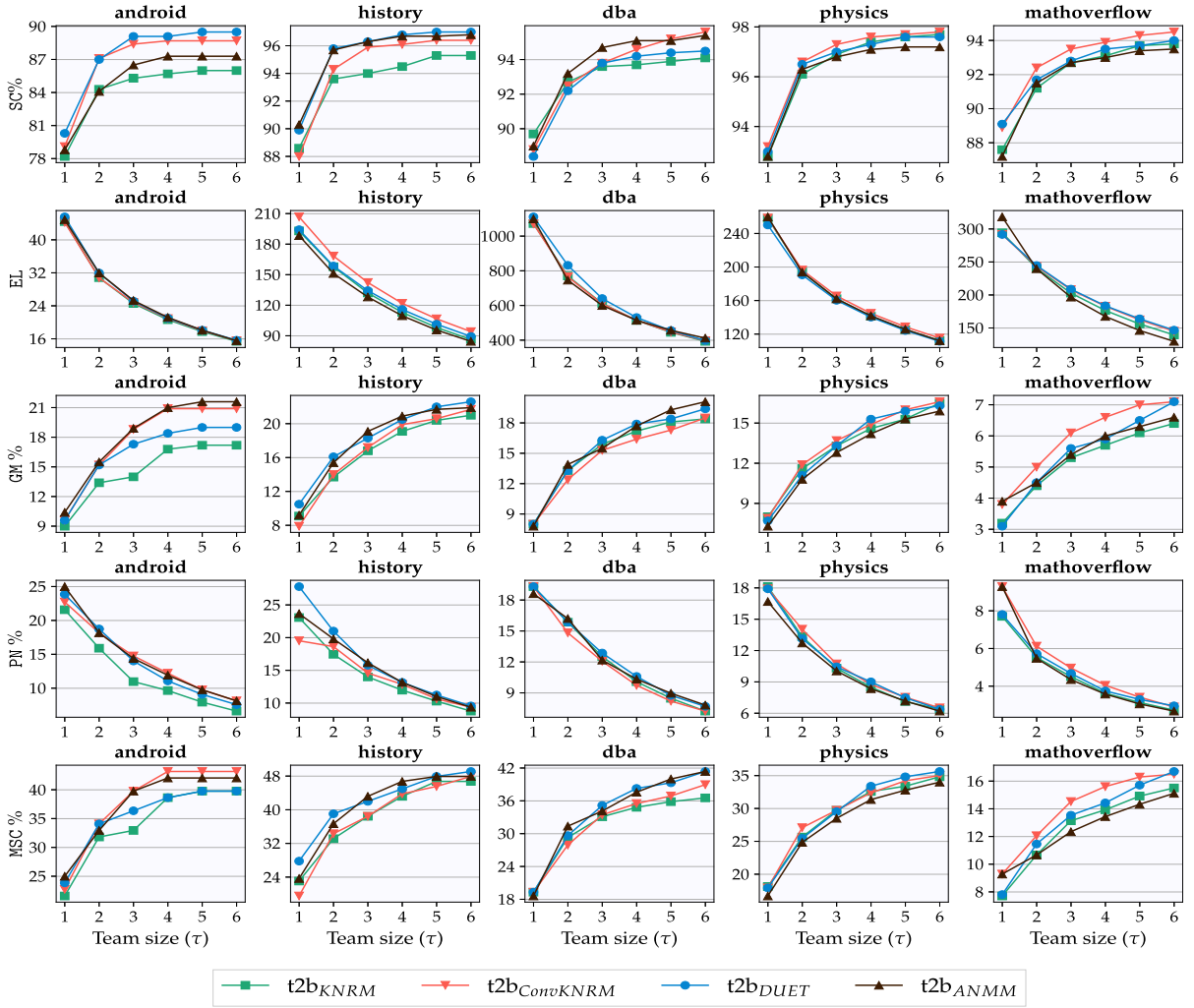

Fig. 8. Impact of using different learn to rank teachings in line 5 of Algorithm 1.

expert given a new question. Our empirical experiments shows that building new teams based on the latent representation of past successful teams leads to members in new teams with high skill coverage, expertise level, and willingness to work as a team to answer the new question. Furthermore, obtaining a higher skill coverage over the baselines indicates that teams formed by team2box possess higher likelihood of having complementary skill sets ($\ell 2$). In addition, team2box addresses $\ell 3$ by learning the embedding of teams along with questions and experts, and employing such latent representations to retrieve new teams. Such an approach results in superior teams in terms of past collaboration level compared to the baselines. The main implications of using team2box are:

- Efficiently exploring the past successful teams to discover a group of suitable experts to address the information needed by a new question. Our proposed subgraph embedding technique preserves the structure of past successful teams as subgraphs of CQA heterogeneous network in the embedding space. Given a new question, members of most relevant teams can be efficiently retrieved in the embedding space as potential answerers to form the new team.
- Utilizing both the textual and structural information in the CQA environments to better connect information seekers with expertise knowledge. The content of existing questions and answers is employed to learn the embedding of words used in the environment. Such latent representations of words are employed to map new questions based on their contents into proper points in the embedding space that preserves the structural information of the CQA environment. Such a technique leads to map a new question close to teams in which they have answered existing questions with similar content to the new question. Hence, members of the closest teams possess a high probability of having background knowledge required to provide the knowledge needed by information seekers.
- Increasing the engagement of experts in the community by grouping them with their past teammates. team2box chooses the members of a new team from experts in the most relevant existing teams to the new question. Such an approach results in a higher likelihood of having members with past cooperation among the members of the new constructed team.

6. Concluding remarks

In this paper, we proposed a novel embedding-based team formation method, called team2box, for forming teams in community question answering platforms. team2box learns a vector space by considering the structure of teams and the relationship between experts and questions. Akin to recent *expert finding* methods, team2box leverages the structural properties of a social network built based on the relationships among experts and questions to improve the quality of retrieved experts by effectively learning the latent representations of questions and experts. Compared to its counterparts, team2box goes one step further and learns the structure of past collaborations of experts to answer common questions as boxes in the same latent space of experts and questions. This results in the reflection of past collaborative relationship between the experts in the learnt embeddings. In addition, team2box utilizes the voting scores of answers while learning the embeddings for experts and questions by building a weighted heterogeneous graphical representation, which is overlooked in the state of the art. This increases the quality of the predicted experts in terms of their level of expertise. We have reported on extensive experiments on real-world datasets from a variety of domains with different sizes. In our experiments, we compared the performance of team2box against two different sets of baselines, namely (1) expert finding baselines, and (2) team formation baselines. The experiments show that our method (a) is more successful for team formation compared to the baselines, and (b) is independent of the dataset and shows a consistent better performance over the baselines regardless of the baseline or the dataset.

As a future work, we will focus more on the structure of discovered teams and the roles needed in each team. As a starting point, team2box considers the actual answerers of a question as a team, and learns a model to discover new teams for new posted questions. However, in expert team recommendation, there are many ways to form a team. An important challenge is which team is more effective than others. Thus, we will expand this work to study the properties of existing effective teams and utilize them to form new teams. In reality, users with different roles such as question reviewers, question editors, answerers, and answer editors with variety of level of expertise work together to answer new posted questions in CQA systems. To form more realistic teams, we will consider such roles in recommended teams by ranking experts based on roles needed in new teams.

CRediT authorship contribution statement

Roohollah Etemadi: Writing - original draft, Methodology, Software, Investigation, Visualization. **Morteza Zihayat:** Supervision, Conceptualization, Methodology, Writing - review & editing, Funding acquisition. **Kuan Feng:** Supervision, Conceptualization, Methodology. **Jason Adelman:** Supervision, Conceptualization, Methodology. **Ebrahim Bagheri:** Supervision, Conceptualization, Methodology, Writing - review & editing, Funding acquisition.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Morteza Zihayat reports financial support was provided by Natural Sciences and Engineering Research

Council of Canada. Ebrahim Bagheri reports financial support was provided by Natural Sciences and Engineering Research Council of Canada.

References

[1] Z. Li, J.-Y. Jiang, Y. Sun, W. Wang, Personalized question routing via heterogeneous network embedding, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 192–199.
[2] J. Sun, J. Zhao, H. Sun, S. Parthasarathy, Endcold: An end-to-end framework for cold question routing in community question answering services, in: Proceedings of IJCAI'20, 2020, pp. 3244–3250.
[3] S. Yuan, Y. Zhang, W. Hall, J.B. Cabotà, Expert finding in community question answering: a review, AI Rev. 53 (2) (2020) 843–874.
[4] X. Zhang, W. Cheng, B. Zong, Y. Chen, J. Xu, D. Li, H. Chen, Temporal context-aware representation learning for question routing, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 753–761.
[5] S. Chang, A. Pal, 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), IEEE, 2013, pp. 494–501.
[6] T. Lappas, K. Liu, E. Terzi, Finding a team of experts in social networks, in: Proceedings of the 15th ACM SIGKDD, 2009, pp. 467–476.
[7] A. Khan, L. Golab, M. Kargar, J. Szlichta, M. Zihayat, Compact group discovery in attributed graphs and social networks, Inform. Process. Manage. 57 (2) (2020) 102054.
[8] Y. Kou, D. Shen, Q. Snell, D. Li, T. Nie, G. Yu, S. Ma, Efficient team formation in social networks based on constrained pattern graph, in: 2020 IEEE 36th International Conference on Data Engineering (ICDE), IEEE, 2020, pp. 889–900.
[9] M. Kargar, L. Golab, D. Srivastava, J. Szlichta, M. Zihayat, Effective keyword search over weighted graphs, IEEE Trans. Knowl. Data Eng.
[10] M. Zihayat, A. An, L. Golab, M. Kargar, J. Szlichta, Authority-based team discovery in social networks, International Conference on Extending Database Technology (EDBT).
[11] G. Zhou, S. Lai, K. Liu, J. Zhao, Topic-sensitive probabilistic model for expert finding in question answer communities, in: Proceedings of the 21st ACM international conference on Information and knowledge management, 2012, pp. 1662–1666.
[12] Z. Zhao, H. Lu, V.W. Zheng, D. Cai, X. He, Y. Zhuang, Community-based question answering via asymmetric multi-faceted ranking network learning., in: AAAI, Vol. 17, 2017, pp. 3532–3539.
[13] Z. Zhao, L. Zhang, X. He, W. Ng, Expert finding for question answering via graph regularized matrix completion, IEEE Trans. Knowl. Data Eng. 27 (4) (2014) 993–1004.
[14] H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang, M. Ester, Community-based question answering via heterogeneous social network learning, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 122–128.
[15] Z. Zhao, Q. Yang, D. Cai, X. He, Y. Zhuang, Expert finding for community-based question answering via ranking metric network learning., in: International Joint Conferences on Artificial Intelligence (IJCAI), Vol. 16, 2016, pp. 3000–3006.
[16] 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, 2005, pp. 315–316.
[17] 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, 2010, pp. 1585–1588.
[18] 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, 2011, pp. 2041–2044.
[19] G. Zhou, K. Liu, J. Zhao, Joint relevance and answer quality learning for question routing in community qa, in: Proceedings of the 21st ACM international conference on Information and knowledge management, 2012, pp. 1492–1496.
[20] Z. Ji, B. Wang, Learning to rank for question routing in community question answering, in: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 2013, pp. 2363–2368.
[21] L.T. Le, C. Shah, Retrieving people: Identifying potential answerers in community question-answering, J. Assoc. Inform. Sci. Technol. 69 (10) (2018) 1246–1258.
[22] X. Liu, S. Ye, X. Li, Y. Luo, Y. Rao, Zhihurank: A topic-sensitive expert finding algorithm in community question answering websites, in: International Conference on Web-Based Learning, Springer, 2015, pp. 165–173.
[23] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, Z. Chen, Cqarank: jointly model topics and expertise in community question answering, in: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 2013, pp. 99–108.
[24] B. Yang, S. Manandhar, Tag-based expert recommendation in community question answering, in: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), IEEE, 2014, pp. 960–963.
[25] S. Sorkhani, R. Etemadi, A. Bigdeli, M. Zihayat, E. Bagheri, Feature-based question routing in community question answering platforms, Inf. Sci. 608 (2022) 696–717.
[26] W. Tang, T. Lu, D. Li, H. Gu, N. Gu, Hierarchical attentional factorization machines for expert recommendation in community question answering, IEEE Access 8 (2020) 35331–35343.
[27] D. Kundu, R.K. Pal, D.P. Mandal, Topic sensitive hybrid expertise retrieval system in community question answering services, Knowl.-Based Syst. 211 (2021) 106535.
[28] J. Fu, Y. Li, Q. Zhang, Q. Wu, R. Ma, X. Huang, Y.-G. Jiang, Recurrent memory reasoning network for expert finding in community question answering, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 187–195.
[29] A. Pal, F. Wang, M.X. Zhou, J. Nichols, B.A. Smith, Question routing to user communities, in: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 2013, pp. 2357–2362.
[30] R.W. White, M. Richardson, Effects of expertise differences in synchronous social q&a, in: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, 2012, pp. 1055–1056.
[31] L.T. Le, C. Shah, Retrieving rising stars in focused community question-answering, in: Asian Conference on Intelligent Information and Database Systems, Springer, 2016, pp. 25–36.
[32] L. Nie, Y. Li, F. Feng, X. Song, M. Wang, Y. Wang, Large-scale question tagging via joint question-topic embedding learning, ACM Trans. Inform. Syst. (TOIS) 38 (2) (2020) 1–23.
[33] Y. Liu, W. Tang, Z. Liu, L. Ding, A. Tang, High-quality domain expert finding method in cqa based on multi-granularity semantic analysis and interest drift, Inf. Sci. 596 (2022) 395–413.
[34] S. Jimenez, F.N. Silva, G. Dueñas, A. Gelbukh, Proficiencyrank: Automatically ranking expertise in online collaborative social networks, Inf. Sci. 588 (2022) 231–247.
[35] M. Kargar, A. An, M. Zihayat, Efficient bi-objective team formation in social networks, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2012, pp. 483–498.
[36] S.M. Nikolakaki, E. Pitoura, E. Terzi, P. Tsaparas, Finding teams of maximum mutual respect, ICDM, IEEE.
[37] J. Huang, Z. Lv, Y. Zhou, H. Li, H. Sun, X. Jia, Forming grouped teams with efficient collaboration in social networks, Comput. J. 60 (11) (2017) 1545–1560.
[38] C.-T. Li, M.-K. Shan, S.-D. Lin, On team formation with expertise query in collaborative social networks, Knowl. Inf. Syst. 42 (2) (2015) 441–463.
[39] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, S. Leonardi, Online team formation in social networks, in: Proceedings of the 21st international conference on World Wide Web, 2012, pp. 839–848.

[40] H. Rahman, S.B. Roy, S. Thirumuruganathan, S. Amer-Yahia, G. Das, Optimized group formation for solving collaborative tasks, VLDB J. 28 (1) (2019) 1–23.
[41] S. Apostolou, P. Tsaparas, E. Terzi, Template-driven team formation, in: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2020.
[42] S. Bahargam, B. Golshan, T. Lappas, E. Terzi, A team-formation algorithm for faultline minimization, Expert Syst. Appl. 119 (2019) 441–455.
[43] Y. Dong, N.V. Chawla, A. Swami, metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of ACM SIGKDD, 2017, pp. 135–144.
[44] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.
[45] L. Xu, X. Wei, J. Cao, P.S. Yu, Embedding of embedding (eoe) joint embedding for coupled heterogeneous networks, in: Proceedings of WSDM, 2017, pp. 741–749.
[46] D. Kingma, J. Lei Ba, A method for stochastic optimization, International Conference on Learning Representations. San Dego: ICLR.
[47] C. Xiong, Z. Dai, J. Callan, Z. Liu, R. Power, End-to-end neural ad-hoc ranking with kernel pooling, in: Proceedings of ACM SIGIR'17, 2017, pp. 55–64.
[48] B. Mitra, F. Diaz, N. Craswell, Learning to match using local and distributed representations of text for web search, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1291–1299.
[49] L. Yang, Q. Ai, J. Guo, W.B. Croft, anmm: Ranking short answer texts with attention-based neural matching model, in: Proceedings of the 25th ACM international on conference on information and knowledge management, 2016, pp. 287–296.
[50] Z. Dai, C. Xiong, J. Callan, Z. Liu, Convolutional neural networks for soft-matching n-grams in ad-hoc search, in: Proceedings of the eleventh ACM international conference on web search and data mining, 2018, pp. 126–134.