Graph Collaborative Expert Finding with Contrastive Learning

Qiyao Peng[1] , **Wenjun Wang***[2,3,5] , **Hongtao Liu**[4] , **Cuiying Huo**[2] and **Minglai Shao***[1]
[1]School of New Media and Communication, Tianjin University, Tianjin, China
[2]College of Intelligence and Computing, Tianjin University, Tianjin, China
[3]Georgia Tech Shenzhen Institute, Tianjin University, Guangdong, China
[4]Du Xiaoman Finicial Technology, Beijing, China
[5]Yazhou Bay Innovation Institute, Hainan Tropical Ocean University, Sanya Hainan, China
{qypeng, wjwang}@tju.edu.cn, liuhongtao01@duxiaoman.com, {huocuiying, shaoml}@tju.edu.cn

Abstract

In Community Question Answering (CQA) websites, most current expert finding methods often model expert embeddings from textual features and optimize them with expert-question first-order interactions, i.e., *this expert has answered this question*. In this paper, we try to address the limitation of current models that typically neglect the intrinsic high-order connectivity within expert-question interactions, which is pivotal for collaborative effects. We introduce an innovative and simple approach: by conceptualizing expert-question interactions as a bipartite graph, and then we propose a novel graph-based expert finding method based on contrastive learning to effectively capture both first-order and intricate high-order connectivity, named CGEF. Specifically, we employ a question encoder to model questions from titles and employ the graph attention network to recursively propagate embeddings. Besides, to alleviate the problem of sparse interactions, we devise two auxiliary tasks to enhance expert modeling. First, we generate multiple views of one expert, including: 1) behavior-level augmentation drops interaction edges randomly in the graph; 2) interest-level augmentation randomly replaces question titles with tags in the graph. Then we maximize the agreement between one expert and the corresponding augmented expert on a specific view. In this way, the model can effectively inject collaborative signals into expert modeling. Extensive experiments on six CQA datasets demonstrate significant improvements compared with recent methods.

1 Introduction

Lots of people have begun to use online Community Question Answering (CQA) platforms, such as StackOverflow, Zhihu for asking questions and seeking helps. To help CQA users efficiently obtain high-quality answers [Zhao *et al.*, 2017; Hu *et al.*, 2021], expert finding technique that aims to finding suitable experts for unanswered questions, is widely used [Zhao *et al.*, 2014; Yuan *et al.*, 2020; Peng *et al.*, 2022a; Amendola *et al.*, 2024].

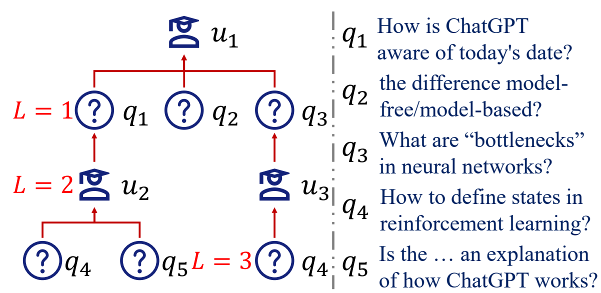

Figure 1: Example of the high-order connectivity (e.g., "$L = 3$").

A key point in expert finding is how to learn better expert representations. Most existing works [Robertson and Zaragoza, 2009; Zhou *et al.*, 2012; Ghasemi *et al.*, 2021; Peng *et al.*, 2022a] utilize various learning methods to automatically learn informative expert representations from their historical behaviors. Some works [Chang and Pal, 2013; Yang *et al.*, 2013; Liu *et al.*, 2015] employ traditional methods to learn question topic features and model experts. Ji et al. [Ji and Wang, 2013] introduce statistical features and intrinsic relationships into a learning to rank model for ranking potential answerers. Furthermore, some works employ the neural networks (e.g., Transformer [Vaswani *et al.*, 2017]) to model experts [Zhang *et al.*, 2020; Peng *et al.*, 2022a]. PMEF [Peng *et al.*, 2022a] uses a multi-view learning method for modeling more comprehensive expert embeddings from the expert-question interactions. These models have achieved effective performance on expert finding.

However, they often model experts based on the first-order interactions (i.e., "L=1" in Figure 1) and omit the high-order connectivity underlying expert-question interactions. In fact, the high-order connectivity could help inject the collaborative effects into the model and improve the quality of user representations [Wang *et al.*, 2019b; Wu *et al.*, 2021]. For example, as illustrated in Figure 1, the path $u_2 \rightarrow q_1 \rightarrow u_1$ indicates the behavior similarity between u_1 and u_2, because they have answered a same question q_1; compared with q_5, q_4 is more likely recommended to u_1, since q_4 has two longer path (e.g., $q_4 \rightarrow u_3 \rightarrow q_3 \rightarrow u_1$) with u_1 while q_5 has only one

* Corresponding author

path. In a word, if the model could not capture this high-order connectivity, it would lose the ability to model the expert u_1 using the question q_4, which results in the model being unable to recommend questions like q_4 to the expert.

Recently, the Graph Neural Networks [Kipf and Welling, 2016] have demonstrated powerful capabilities in various field, e.g., recommender systems [Wang *et al.*, 2019b; Wu *et al.*, 2021]. This inspires us to construct an expert-question interaction graph and employ the GNN-based method for capturing high-order connectivity. However, it is non-trivial to employ the GNN-based method in expert finding due to the challenge of *sparse expert answering behaviors*. For example, in English dataset from StackExchange, the ratio of interaction accounts for 0.047% in the all interaction space. Moreover, the power-law distribution commonly observed in the interactions among expert-questions. However, the GNN-based methods would easily bias towards high-degree experts, making it insufficient to model low-degree experts.

In this paper, inspired by [Wang *et al.*, 2019b], we propose a Contrastive Graph learning model for Expert Finding (CGEF), aiming to effectively capture high-order connectivity underlying the expert-question interactions. Towards this end, we construct an expert-question bipartite graph based on the historical interactions and propagate expert and question representations along the graph. Considering the target question is usually a new question during inference, we employ the question title and design a question encoder for initializing question embeddings as the question node information. During propagating, since different neighbors or different high-order paths have different importance on modeling experts, the graph attention network [Veličković *et al.*, 2018] is utilized to obtain the weight of each neighbor for modeling experts. Besides, for enhancing expert modeling with limited interaction data, we construct auxiliary tasks for discriminating the expert representation of an expert itself. Specifically, it contains: *1) data augmentation* to generate different views of the graph. We design a behavior-level augmentation to randomly drop interactions in the graph, which might help capture the more useful interaction patterns. And we propose an interest-level augmentation to randomly replace question titles as question tags for helping learn different grained expert interests; *2) contrastive learning* to promote consistent representations across different perspectives of the same expert on the modified graph. During the inference phase, we use the candidate expert embedding and the target question embedding (derived from the question encoder) to match. The contributions are summarized as:

(1) To the best of our knowledge, we are the first to propose the GNN-based contrastive expert finding model in the CQA, which can effectively capture high-order connectivity underlying expert-question interactions.

(2) For alleviating the sparsity of expert-question interaction, we design two auxiliary tasks based on the behavior-level and interest-level to offer additional supervised signals for expert representation learning.

(3) Extensive experimental results demonstrate that CGEF outperforms existing baselines, validating the effectiveness of capturing high-order connectivity.

https://archive.org/details/stackexchange

2 Related Work

2.1 Expert Finding

Expert finding is to predict whether a CQA user will provide a suitable answer to a given question [Li and King, 2010]. Many of these approaches utilize neural networks (e.g., convolutional neural network), to acquire question or expert characteristics through the modeling of question-expert interactions and question content. For example, RMRN [Fu *et al.*, 2020] introduced a novel recurrent memory reasoning mechanism to address the challenge of matching question semantics by learning the implicit relevance between the expert and target questions. Peng et al. [Peng *et al.*, 2022b] designed a hierarchical matching network to learn multi-grained and comprehensive matching clues for finding experts better. MATER [Zahedi *et al.*, 2024] considered both time-awareness user interest and expertise for expert finding, and have obtained superior performance. Nevertheless, these methods might omit the higher-order connectivity for modeling experts. Another line of research [Li *et al.*, 2019; Ghasemi *et al.*, 2021] exploited the expert-question graph to infer expert preference. For instance, NeRank [Li *et al.*, 2019] constructed a CQA heterogeneous information network with a metapath2vec [Dong *et al.*, 2017] method for learning answerer interests and routed new questions to high-ranking answerers. Ghasemi et al. [Ghasemi *et al.*, 2021] proposed a framework to simultaneously capture semantic similarities from question-answer and use node2vec to capture experts' relations based on graphs, which could benefit finding experts. However, although these methods have achieved good results, they only could model the co-occurrence probability of nodes.

2.2 Graph Neural Recommendation

In recent times, Graph Neural Networks have demonstrated remarkable performance in acquiring node embeddings by leveraging the amalgamation of node features and the inherent graph structure [Huo *et al.*, 2023a; Huo *et al.*, 2023b]. During the propagation process, it iteratively gathers information from neighboring nodes and integrates this aggregated information with the central node embedding [Wu *et al.*, 2022], which have been applied to diverse domains [Battaglia *et al.*, 2016; Wang *et al.*, 2019a; Wang *et al.*, 2019b; Wu *et al.*, 2021]. For example, GC-MC [van den Berg *et al.*, 2017] utilized a graph auto-encoder framework to complete the user-item interaction matrix. GraphRec [Fan *et al.*, 2019] employed a social-based graph recommender model to capture both interactions and opinions, enabling more accurate recommendations. These methods all demonstrate the powerful capability of the GNNs in different recommendation domains. However, different from these methods that rely on user and item IDs, expert finding has its own unique characteristics, e.g., the target question is usually a new question during inference, and the model could not infer the new question representation based on the existing GNN methods.

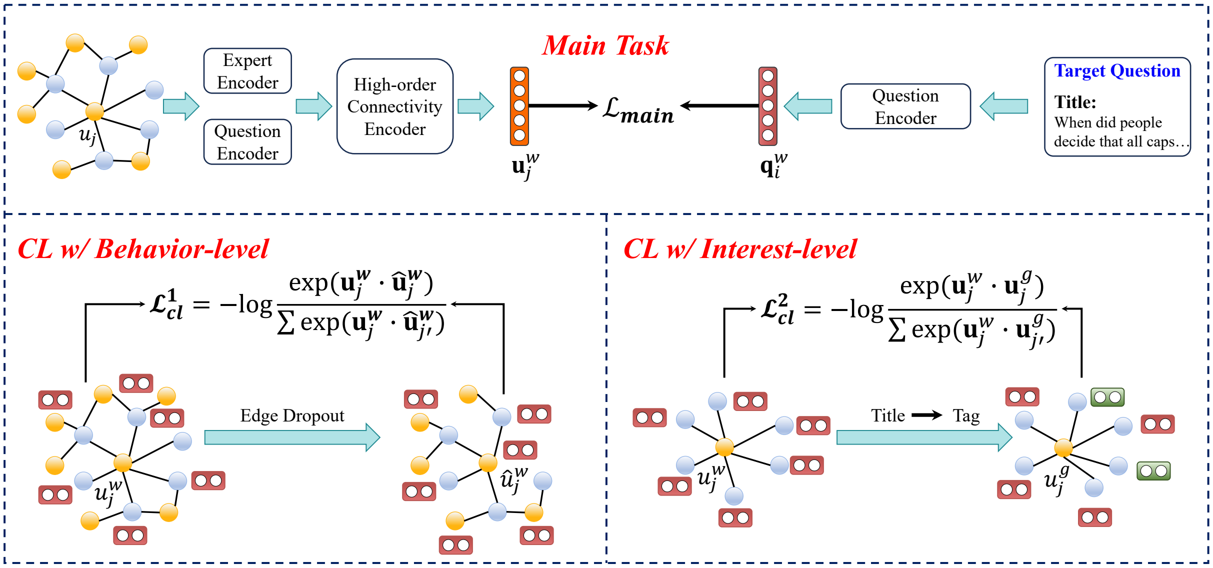

Figure 2: This demonstrates the main process of our proposed model, CGEF. The top layer illustrates the workflow of the primary supervised expert finding task. The bottom layer represents the operational processes of contrastive learning introduced within the model.

3 Method

We assume an expert set is $U = \{u_1, u_2, \cdots, u_m\}$ containing m experts and a question set is $Q = \{q_1, q_2, \cdots, q_n\}$ containing n questions. For the i-th question q_i in Q, it is associated with the tag set $q_i^g = \{g_1^i, g_2^i, \cdots, g_v^i\}$ including v tags, and the title word set $q_i^w = \{w_1^i, w_2^i, \cdots, w_l^i\}$ including l words. The tag and title of the target question q^t are same. It is noted that the expert who provides the "accepted answer" is the ground truth and the objective of the paper is to predict the most suitable expert, delivering an "accepted answer" for a given question. Then, we introduce CGEF, which can effectively capture expert-question high-order collaborative information for modeling experts. The working flow of CGEF are shown in Figure 2.

3.1 Interaction Graph Construction

In this part, based on the interactions, we construct the expert-question interaction graph, where the expert and question embeddings are derived from the expert and question encoder.

Expert Encoder

We first embed each expert as a vector for initializing expert node embedding. Specifically, we prepare an embedding lookup table $\mathbf{U} \in \mathcal{R}^{m \times d}$, where m represents the number of experts. Afterward, we embed each expert. For the i-th expert, the embedding is retrieved by:

$$\mathbf{u}_i = \mathbf{U}h(i), \tag{1}$$

where $h(i)$ is a one-hot vector with m as dimension, and the non-zero element is the position of i in expert's embedding $\mathbf{U}$. The embedding table serves as initializing expert node embeddings and will be optimized during the model training.

Question Encoder

In the expert finding, the target question is usually a new question during inference, i.e., this question usually has not been answered before. Hence, we could not directly use the question ID to embed the new target question like expert, which makes it impossible to calculate the matching score. Fortunately, the question has the off-the-shell question title information, which could be employed for inferring the question embedding.

Specifically, given the question title $q^w = \{w_1, \cdots, w_l\}$ with l words, we employ the BERT [Devlin *et al.*, 2018] to capture the question overall feature. We include a special token $[CLS]$ at the beginning of q^w and $[SEP]$ at the end, and the input is denoted as follows:

$$\{[CLS], w_1, w_2, \cdots, w_l, [SEP]\}. \tag{2}$$

The output embedding derived from the token $[CLS]$ can be viewed as the question node embedding $\mathbf{q} \in \mathcal{R}^{d_w}$. Then we employ a linear layer to transform the dimension of that as $\mathcal{R}^d$. Note that the parameters of the question encoder will be updated during training, and we employ that to reason the new question representations during inference.

Graph Construction

Then, we construct a bipartite graph $\mathcal{G} = \{\mathcal{U}, \mathcal{Q}, \mathcal{E}\}$ based on the expert-question interactions. $\mathcal{E}$ is the set of edges in the graph $\mathcal{G}$ and each edge $e = (u, q) \in \mathcal{E}$ indicates that expert u has answered question q. The expert node embedding is initialized from the expert encoder (i.e., i-th expert's embedding is $\mathbf{u}_i$) and the question node embedding is initialized with the question encoder (i.e., $\mathbf{q}$). By incorporating the question title information into the expert-question interaction graph, we could alleviate the problem faced by GNN-based methods where new questions cannot be encoded due to the absence of interaction with experts.

3.2 High-order Connectivity Encoder

As denoted above, the high-order connectivity like $q_4 \rightarrow u_3 \rightarrow q_3 \rightarrow u_1$ is crucial to estimating the relevance score between the experts and questions. Based on the graph, we employ the GNN-based method to aggregate neighbor question node information for learning expert embeddings. Besides, different questions and different high-order connectivity paths that the central expert has interacted with are of different informativeness in learning expert embeddings. Hence, we exploit the idea of the graph attention network [Veličković *et al.*, 2018] to capture high-order connectivity for different experts by attentively propagating the embeddings of expert and question nodes on the graph.

Specifically, let $\mathbf{u}^{(l-1)}, \mathbf{q}_j^{(l-1)} \in \mathcal{R}^d$ denote the projected embeddings of the central expert and neighborhood question j after $(l-1)$-th propagation layers. The expert embedding in l-th layer is calculated as follow:

$$\mathbf{u}^{(l)} = \sum_{j=1}^{N} \alpha_{q_j} \mathbf{q}_j^{(l-1)} \ , \tag{3}$$

where N is the neighborhood question node number, and α_{q_j} is the attention weight indicating different importance of questions, which is calculated as follows:

$$\alpha_{q_j}^* = \mathbf{w}_2 \odot [\mathbf{u}^{(l-1)}; \mathbf{q}_j^{(l-1)}] \ , \tag{4}$$

$$\alpha_{q_j} = \frac{\exp(\text{LeakyReLU}(\alpha_{q_j}^*))}{\sum_{j=1}^{N} \exp(\text{LeakyReLU}(\alpha_{q_j}^*))} \ , \tag{5}$$

where the $[\cdot;\cdot]$ operator is the concatenate operator, and $\text{LeakyReLU}(\cdot)$ is the activation function. The final expert and question embeddings are denoted as $\mathbf{u}^w$ and $\mathbf{q}^w$.

Next, we introduce the traditional expert finding training loss, which is built upon the final expert and question embeddings to predict expert-question matching score. A classical solution to predict the score S_c can be denoted as follows:

$$S_c = (\mathbf{u}^w)^T \mathbf{q}^w \ . \tag{6}$$

The way to optimize the parameters in most existing method can be denoted as follows:

$$\bar{S}_c = \frac{S_c}{\sum_{g=1}^{K+1} \exp(S_g)}, \ c \in \{1, 2, \cdots, K+1\}, \tag{7}$$

$$\mathcal{L}_{main} = -\sum_{c=1}^{K+1} \hat{S}_c log \bar{S}_c, \tag{8}$$

where K is the number of negative samples, and $\hat{S}_c$ denotes the ground truth, and S_c represents the predicted probability, and $\bar{S}_c$ denotes the normalized probability. Note that during the inference, the target questions are usually not answered by any experts. Their embeddings are generated from the question encoder without neighbor aggregation. In our method, we choose it as the primary supervised task.

3.3 Expert Contrastive Learning

In this section, we introduce two auxiliary tasks aimed at mitigating the issue of sparse historical behaviors.

1) Behavior-level. The expert representation learning process may be affected by the limited interaction behaviors exhibited in most experts. Besides, these low-degree experts would be greatly influenced by the high-degree (with a large number of interactions) experts during the representation learning. In this part, we devise a behavior-level contrastive learning task, aiming to capture more valuable interaction patterns within the expert-specific local structure, which could help mitigate the impact of high-degree expert nodes.

Specifically, we randomly drop interactions in the original graph with a dropout ratio ρ as an augmentation view. For the j-th expert, we employ the above encoder to obtain original embedding $\mathbf{u}_j^w$ and the augmented embedding $\hat{\mathbf{u}}_j^w$ derived from the augmentation graph. The contrastive loss is:

$$\mathcal{L}_{cl}^1 = -\sum_{j=1}^{m} \log \frac{\exp(\text{sim}(\hat{\mathbf{u}}_j^w \cdot \mathbf{u}_j^w)/\tau)}{\sum_{j'=1}^{m} \exp(\text{sim}(\hat{\mathbf{u}}_j^w \cdot \mathbf{u}_{j'}^w)/\tau)} \ , \tag{9}$$

where $\text{sim}(\cdot)$ quantifies the resemblance between two vectors, τ is a temperature parameter.

2) Interest-level. On many CQA websites, the question is usually labeled with some tags, which are highly related to the question field and reflect the expert's coarse-grained interests. We directly introduce the question tag as a natural augmented information to construct an augmented graph. This allows us to distinguish the representation of a particular expert from other experts, while capturing both the different grained interests of the experts.

Specifically, given the v-th tag g_v in the i-th question tag set, we embed the v-th tag as a low-dimensional feature vector as $\mathbf{g}_v \in \mathcal{R}^d$ via a tag embedding layer. Then, we aggregate all tag features of this question with a max aggregator, which is computed as follow:

$$\mathbf{q}_i^g = \max[\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_v], \tag{10}$$

where $\mathbf{q}_i^g \in \mathcal{R}^d$ is the i-th question tag embedding.

Then, we randomly replace the question title embedding $\mathbf{q}^w$ as the question tag embedding $\mathbf{q}^g$ with the ratio λ in the graph. Based on this, given the j-th expert, we can obtain the original embedding $\mathbf{u}_j^w$ and the augmented embedding $\mathbf{u}_j^g$. Afterwards, the augmented embedding $\mathbf{u}_j^g$ is considered as the positive one of $\mathbf{u}_j^w$, while other experts are considered as negative samples, and the contrastive loss is:

$$\mathcal{L}_{cl}^2 = -\sum_{j=1}^{m} \log \frac{\exp(\text{sim}(\mathbf{u}_j^w \cdot \mathbf{u}_j^g)/\tau)}{\sum_{j'=1}^{m} \exp(\text{sim}(\mathbf{u}_j^w \cdot \mathbf{u}_{j'}^g)/\tau)} \ . \tag{11}$$

3.4 Multi-task Training

To improve expert finding with the above auxiliary tasks, we leverage a multi-task training strategy with α as the controller parameter to jointly unify these tasks, which is defined as:

$$\mathcal{L} = \mathcal{L}_{main} + \alpha(\mathcal{L}_{cl}^1 + \mathcal{L}_{cl}^2) \ . \tag{12}$$

4 Experiment

4.1 Experimental Settings

Our experiments are conducted on real-world CQA datasets from StackExchange. Following previous works, we utilize

https://archive.org/details/stackexchange

Dateset	*AI*			*Bioinformatics*			*Print*		
Metric / Method	MRR	P@1	NDCG@10	MRR	P@1	NDCG@10	MRR	P@1	NDCG@10
CNTN	.4516	.2777	.5153	.4369	.2551	.5228	.4762	.2918	.5532
NeRank	.4989	.3313	.5407	.4515	.2812	.5266	.5136	.3135	.6058
TCQR	.4206	.2828	.4916	.3956	.2611	.4451	.4521	.2677	.5250
RMRN	.4526	.2789	.5139	.4446	.2728	.5312	.4836	.3019	.5646
UserEmb	.4086	.2405	.4687	.3799	.2515	.4319	.3628	.2865	.5053
PMEF	.5226	.3509	.5588	.4666	.3089	.5405	.5212	.3426	.6161
EFHM	.5135	.3343	.5623	.4603	.3055	.5371	.5236	.3627	.6063
CGEF	**.5443**	**.3889**	**.5846**	**.4723**	**.3544**	**.5592**	**.5532**	**.4043**	**.6225**
Dataset	*History*			*English*			*Biology*		
Metric / Method	MRR	P@1	NDCG@10	MRR	P@1	NDCG@10	MRR	P@1	NDCG@10
CNTN	.3940	.2546	.4332	.2988	.1857	.4231	.3368	.2019	.3692
NeRank	.4685	.2783	.5653	.4896	.2819	.5643	.4231	.2389	.4990
TCQR	.4121	.2757	.4966	.3435	.1938	.4826	.4062	.2436	.4756
RMRN	.5308	.3356	.6136	.4680	.2632	.5685	.4386	.2463	.5188
UserEmb	.4169	.2739	.4463	.3163	.1966	.4652	.3323	.2068	.3626
PMEF	.5533	.3677	.6407	.4967	.2925	.5885	.4720	.2962	.5508
EFHM	.5686	.3688	.6401	.5077	.3186	.6099	.4850	.3053	.5229
CGEF	**.6029**	**.4534**	**.6664**	**.5098**	**.3412**	**.6111**	**.4901**	**.3346**	**.5670**

Table 1: Expert finding results of different baselines (underlined is the best performance in baselines).

Statistics / Datasets	*# questions*	*# answerers*	*# answers*	*avg.title length*
AI	1,205	195	1,719	10.97
Print	1,033	112	1,686	9.51
History	4,904	471	9,452	12.38
Biology	8,704	630	11,411	9.84
English	46,692	4,781	104,453	9.68
Bioinformatics	958	113	1,489	9.93

Table 2: Statistical details of the datasets.

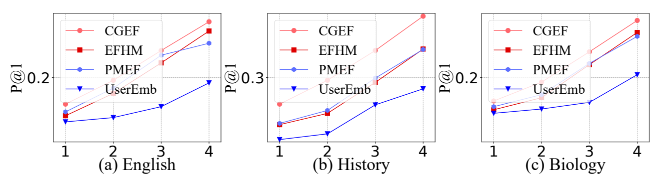

Figure 3: Performance with the different interaction sparsity.

six datasets, i.e., *Bioinformatics*, *English*, *Print*, *History*, *Biology* and *AI* for evaluation. Every dataset consists of a set of questions, where each question is accompanied by its title, tag, and corresponding answerers. Noted that each question has one expert providing the "accepted answers". We preprocess each dataset following the previous works [Li *et al.*, 2019; Zahedi *et al.*, 2024]. For every question, we create a candidate expert set comprising 20 experts. This set includes the original answerer, who provided the accepted answer, along with other randomly selected experts from the answerer pool. The word length of the question title is 15. Table 2 summarizes comprehensive statistics for six datasets in detail. We exclude other CQA datasets, like Yahoo! Answers, from our evaluation as they do not provide the necessary "accepted answers" that serve as the ground truth [Li *et al.*, 2019]. Each dataset is partitioned into three distinct sets, namely a training set, a validation set, and a testing set. The allocation ratios for these sets are 80%, 10%, and 10% respectively, maintaining the chronological order. To evaluate the ranking quality, we utilize commonly employed recommendation ranking metrics, including Mean Reciprocal Rank (MRR), Precision@1 (P@1), and Normalized Discounted Cumulative Gain (NDCG@10).

The hyperparameters were tuned using the validation set. The dimensions of the question and expert embeddings (d) were set to 100. The embedding dimension in the high-order connectivity encoder is set to 384. The model consisted of 3 graph attention layers, and the batch size of 128 was used. To mitigate overfitting, a dropout technique [Srivastava *et al.*, 2014] was employed with a dropout ratio of 0.3. We employ the Adam [Kingma and Ba, 2015] to optimize our model, setting the learning rate to 0.001 and the weight decay to 0.0005. The interest-level replacing ratio λ is 0.3 and the behavior-level dropping ratio ρ is 0.25. The temperature τ is 0.1.

4.2 Performance-Overall Evaluation

We evaluate the performance of our method, *CGEF*, by comparing it with several recent competitive approaches, including: **(1) CNTN** [Qiu and Huang, 2015]: It uses convolutional neural networks for learning question embeddings; **(2) NeRank** [Li *et al.*, 2019]: NeRank learns question content/raiser/answerers embeddings, then uses a scoring function for routing questions; **(3) TCQR** [Zhang *et al.*, 2020]: This method uses a temporal context-aware model for question routing that employs a temporal-aware attention with multi-shift/resolution functions; **(4) RMRN** [Fu *et al.*, 2020]: For exploring the implicit relevance between experts and questions, it leverages a novel recurrent memory reasoning network for expert finding; **(5) UserEmb** [Ghasemi *et al.*, 2021]: It combines experts' community relations and question-answer semantic relationships for finding experts; **(6) PMEF** [Peng *et al.*, 2022a]: This paper proposes an attentive multi-view mechanism for expert finding, which could comprehensively learn different grained expert-question relationships; **(7) EFHM** [Peng *et al.*, 2022b]: EFHM uses a multi-grained hierarchical matching mechanism to model three-tier semantic matching information for expert finding. For models that have been open sourced (e.g, NeRank, PMEF, etc.), we directly use the public code for evaluation. For models that are not open sourced (e.g., RMRN, UserEmb, etc.), we reproduce them according to their original paper for evaluation. We conduct each experiment independently, repeating it 5 times, and report the average results.

Table 1 shows the performance w.r.t. ranking performance among the methods on six CQA datasets. We can see: (1) The approaches that take into account the multi-grained interests of experts (e.g., PMEF) typically outperform those that disregard them (e.g., CNTN, TCQR, UserEmb). This is because user interests are multi-grained. It is difficult for a single representation vector to model user interests, which may be sub-optimal for expert finding; (2) Our CGEF consistently outperforms other baseline approaches. It is noted that, on the difficult P@1 metric, CGEF improves the best deep neural model EFHM more than 22.9% on the History dataset. This phenomenon could be attributed to our approach's ability to capture different-grained high-order connectivity information from the expert-question interaction graph, thereby enhancing expert modeling.

4.3 Performance-Interaction Sparsity Level

For experts with little historical behaviors, capturing high-order connectivity could help model the expert's interests. In this section, we conduct experiments on different expert groups with varying levels of behavior sparsity to explore the effectiveness of the high-order connectivity. We select experts randomly and divide these into four different groups according to the number of interactions per expert. Taking the *English*, *History*, and *Biology* datasets as examples, the larger the group is, the larger interaction numbers the experts have. Figure 3 illustrates the results with respect to P@1 on various expert groups in different datasets. We can see that CGEF consistently outperforms the other methods. It demonstrates the importance of capturing high-order connectivity, which include both the first-order interactions learned by most baselines and capturing higher-order expert/question information through recursive embedding propagation. Hence, our model has the potential to improve the representation learning of inactive experts.

4.4 Study of CGEF

Effect of Different Contrastive Tasks

In our method, we design two auxiliary tasks for supplementing the supervised expert finding. We explore the behavior-level and interest-level tasks on the model performance in this section. *w/o BL* and *w/o IL* remove the behavior-level task and interest-level task from the model respectively. *w/o All* represents removing all auxiliary tasks. The experimental results are shown in Table 3 and we have the following findings:

Dataset	*Print*		*AI*	
Method \ Metric	MRR	P@1	MRR	P@1
w/o IL	.5388	.3812	.5256	.3611
w/o BL	.5456	.3889	.5320	.3767
w/o All	.5297	.3734	.5177	.3558
CGEF	**.5532**	**.4043**	**.5443**	**.3889**

Table 3: Effect of different contrastive tasks.

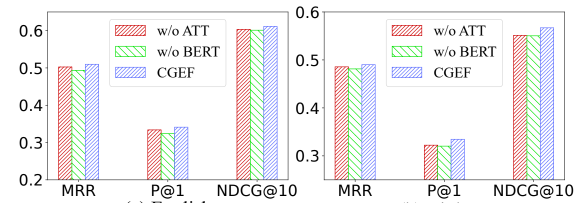

Figure 4: Effect of different modules.

(1) *w/o All* obtains the worst performance. One possible reason for this could be that the supervision signals in expert-question interactions are insufficiently dense to effectively guide the learning of expert representations (e.g. in AI dataset, the interaction is 0.7% compared with the interaction space). These results demonstrate the importance of our designed contrastive learning tasks; (2) *w/o BL* consistently outperforms *w/o IL*. The reason may be that supervision signals of the interest-level task are generated by available question tag information and avoid changing the original graph structure. The behavior-level task needs to randomly drop some edges from the graph, which would dramatically change the graph structure when modeling experts.

Effect of Different Modules

To explore the impact of various modules on model performance, we conduct ablation studies by designing two variations . Specifically, 1) w/o ATT: we replace the α_{q_j} as $1/|N_u|$, where $|N_u|$ is the degree of the expert node u; 2) w/o BERT: we replace the BERT model with the word2vec [Mikolov *et al.*, 2013] to explore the role that the BERT model plays.

The results are shown in Figure 4. We can find *w/o ATT* and *w/o BERT* degrade the model performance. This result highlights the significance of employing an attention mechanism to differentiate the validity of neighboring nodes while learning the representation of the central expert. Additionally, it showcases the effectiveness of BERT in learning question representations.

Effect of Different Propagation Layer Numbers

In fact, the more layer numbers, the higher-order connectivity could be captured. To investigate the influence of different propagation layer numbers, we vary the number of layers within the range of $\{1, 2, 3, 4\}$. Figure 5 shows the

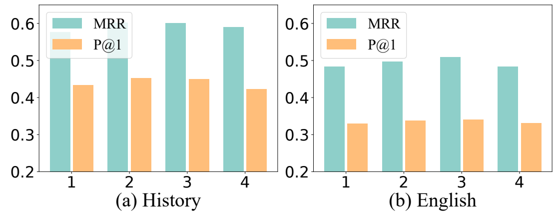

Figure 5: Effect of different number of propagation layers.

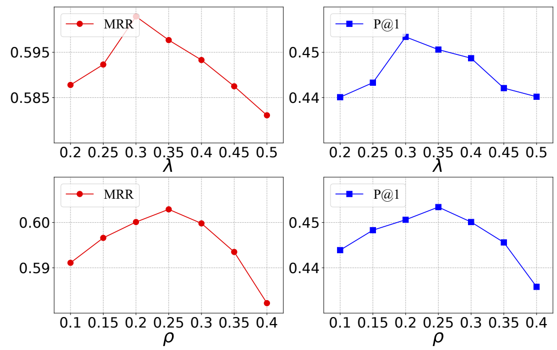

Figure 6: Hyper-parameter analysis.

experimental results. We can see: (1) Increasing the depth of CGEF improves the expert finding performance. *CGEF-2* and *CGEF-3* consistently outperform *CGEF-1* in different datasets, since *CGEF-1* considers only the first-order neighbors. The improvement can be attributed to the effective expert modeling with the second-order and even third-order connectivity effect, which could be captured by the high-order connectivity encoder; (2) Further stacking the layers (e.g., *CGEF-3*) might degrade model performance. The reason may be that applying too deep architecture might induce over-smooth and introduce noises to the expert embedding. Hence, we set the propagation layers as 2 or 3 for different datasets carefully.

4.5 Hyper-parameter Analysis

We explore two important hyper-parameters, including λ, which controls the ratio of title replacing in the graph, and ρ, which controls the ratio of edge dropout in the graph. The experiments are conducted on the History dataset and results are shown in Figure 6. In all, the trend of the impact of the two parameters on the model performance is consistent, both of which increase and then decrease with the increase of parameters. For the λ, introducing fewer (e.g., 30%) question tags to replace question titles could enhance the model's ability to capture expert interests, but excessive replacement would introduce noise into the model. Hence, from the results, we set λ to 0.3 carefully. For the ρ, though the edge dropout is more likely to block connections of high-degree nodes and alleviate the impact of high-degree nodes, the larger ρ would decrease the model performance. The reason is that the larger ρ would dramatically change the graph. Therefore, from the results, we set ρ to 0.25 carefully.

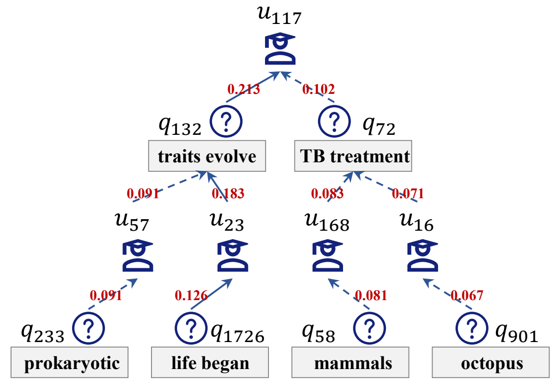

Figure 7: Case Study.

4.6 Case Study

In this part, we perform a case study to gain deeper insights into the what has been learned by the model. We randomly select one expert u_{117} from the Biology dataset and one target question "Does . . . contradict an origin of life?" (from the testing dataset). We extract the expert u_{117} partial high-order connectivity paths in the expert-question graph and the corresponding attention scores. Figure 7 shows the visualization of different high-order path scores. Given the target question, the path "life began $\rightarrow u_{23} \rightarrow$ traits evolve $\rightarrow u_{117}$" can help guide the model to recommend the expert u_{117} to answer the question. Meanwhile, the path could be viewed as evidence of why the expert has the potential to answer this question. Nevertheless, most existing expert finding methods could not capture this kind of high-order collaborative information, which might not match the expert u_{117} with the question precisely. Hence, we can find that the weighted high-order connectivity plays a key role in inferring expert interests.

5 Conclusion

In this paper, we explicitly incorporate high-order connectivity into the expert modeling. In particular, we employ the question encoder to learn the question node embedding for constructing the graph, and then utilize the attention network to capture different importance of different connectivity. Besides, we recognize the challenges of employing the GNN-based method in expert finding and explore corresponding solutions. Specifically, for generating self-supervised signals, we design two data augmentations from behavior-level and interest-level to construct the augmented interaction graph. Based on this, we employ two contrastive learning tasks to supplement the supervised expert finding. In this way, the model can inject collaborative effect into expert modeling. We conduct extensive experiments on six real-world CQA datasets, validating the advantages of our proposed method.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.62272338), and the Shenzhen Sustainable Development Project under Grant (KCXFZ20201221173013036).

References

[Amendola *et al.*, 2024] Maddalena Amendola, Andrea Passarella, and Raffaele Perego. Towards robust expert finding in community question answering platforms. In *Advances in European Conference on Information Retrieval*, volume 14612, pages 152–168. Springer, 2024.

[Battaglia *et al.*, 2016] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, volume 29, 2016.

[Chang and Pal, 2013] Shuo Chang and Aditya Pal. Routing questions for collaborative answering in community question answering. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, pages 494–501, Niagara Ontario, Canada, 2013. ACM.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the International Conference on North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, New Orleans, 2018. Association for Computational Linguistics.

[Dong *et al.*, 2017] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

[Fan *et al.*, 2019] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, pages 417–426. ACM, 2019.

[Fu *et al.*, 2020] Jinlan Fu, Yi Li, Qi Zhang, Qinzhuo Wu, Renfeng Ma, Xuanjing Huang, and Yu-Gang Jiang. Recurrent memory reasoning network for expert finding in community question answering. In *WSDM*, pages 187–195, Houston, Texas, USA, 2020. ACM.

[Ghasemi *et al.*, 2021] Negin Ghasemi, Ramin Fatourechi, and Saeedeh Momtazi. User embedding for expert finding in community question answering. *Proceedings of the ACM Transactions on Knowledge Discovery from Data*, 15(4):1–16, 2021.

[Hu *et al.*, 2021] Jun Hu, Shengsheng Qian, Quan Fang, and Changsheng Xu. Heterogeneous community question answering via social-aware multi-modal co-attention convolutional matching. *IEEE Transactions on Multimedia.*, 23:2321–2334, 2021.

[Huo *et al.*, 2023a] Cuiying Huo, Dongxiao He, Chundong Liang, Di Jin, Tie Qiu, and Lingfei Wu. Trustgnn: Graph neural network-based trust evaluation via learnable propagative and composable nature. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[Huo *et al.*, 2023b] Cuiying Huo, Di Jin, Yawen Li, Dongxiao He, Yu-Bin Yang, and Lingfei Wu. T2-gnn: Graph neural networks for graphs with incomplete features and structure via teacher-student distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4339–4346, 2023.

[Ji and Wang, 2013] Zongcheng Ji and Bin Wang. Learning to rank for question routing in community question answering. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, pages 2363–2368, San Francisco, California, USA, 2013. ACM.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, 2015. OpenReview.net.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Li and King, 2010] Baichuan Li and Irwin King. Routing questions to appropriate answerers in community question answering services. In *Proceedings of the ACM International Conference on Information and knowledge management*, pages 1585–1588, 2010.

[Li *et al.*, 2019] Zeyu Li, Jyun-Yu Jiang, Yizhou Sun, and Wei Wang. Personalized question routing via heterogeneous network embedding. In *Proceedings of the International Conference on Artificial Intelligence*, pages 192–199, Honolulu, Hawaii, USA, 2019. International Joint Conferences on Artificial Intelligence Organization.

[Liu *et al.*, 2015] Xuebo Liu, Shuang Ye, Xin Li, Yonghao Luo, and Yanghui Rao. Zhihurank: A topic-sensitive expert finding algorithm in community question answering websites. In *Proceedings of the International Conference on Web-Based Learning*, pages 165–173. Springer, 2015.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Peng *et al.*, 2022a] Qiyao Peng, Hongtao Liu, Yinghui Wang, Hongyan Xu, Pengfei Jiao, Minglai Shao, and Wenjun Wang. Towards a multi-view attentive matching for personalized expert finding. In *Proceedings of the ACM Web Conference 2022*, pages 2131–2140, Lyon, France, 2022. ACM.

[Peng *et al.*, 2022b] Qiyao Peng, Wenjun Wang, Hongtao Liu, Yinghui Wang, Hongyan Xu, and Minglai Shao. Towards comprehensive expert finding with a hierarchical matching network. *Knowledge-Based Systems*, 257:109933, 2022.

[Qiu and Huang, 2015] Xipeng Qiu and Xuanjing Huang. Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1305–1311, Hyatt Regency in Austin, Texas, USA, 2015. AAAI.

[Robertson and Zaragoza, 2009] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[van den Berg *et al.*, 2017] Rianne van den Berg, Thomas N. Kipf, and Max Welling. Graph convolutional matrix completion. *CoRR*, abs/1706.02263, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the International Conference of Neural Information Processing Systems*, pages 5998–6008, Long Beach, CA, USA, 2017. OpenReview.net.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wang *et al.*, 2019a] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019.

[Wang *et al.*, 2019b] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

[Wu *et al.*, 2021] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021.

[Wu *et al.*, 2022] Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Xiaojie Guo. Graph neural networks: Foundation, frontiers and applications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 4840–4841, New York, NY, USA, 2022. Association for Computing Machinery.

[Yang *et al.*, 2013] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. Cqarank: jointly model topics and expertise in community question answering. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, pages 99–108, San Francisco, California, USA, 2013. ACM.

[Yuan *et al.*, 2020] Sha Yuan, Yu Zhang, Jie Tang, Wendy Hall, and Juan Bautista Cabotà. Expert finding in community question answering: a review. *Artificial Intelligence Review*, 53(2):843–874, 2020.

[Zahedi *et al.*, 2024] Mohammad Sadegh Zahedi, Maseud Rahgozar, and Reza Aghaeizadeh Zoroofi. Mater: Bi-level matching-aggregation model for time-aware expert recommendation. *Expert Systems with Applications*, 237:121576, 2024.

[Zhang *et al.*, 2020] Xuchao Zhang, Wei Cheng, Bo Zong, Yuncong Chen, Jianwu Xu, Ding Li, and Haifeng Chen. Temporal context-aware representation learning for question routing. In *WSDM*, pages 753–761, Houston, Texas, USA, 2020. ACM.

[Zhao *et al.*, 2014] Zhou Zhao, Lijun Zhang, Xiaofei He, and Wilfred Ng. Expert finding for question answering via graph regularized matrix completion. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):993–1004, 2014.

[Zhao *et al.*, 2017] Zhou Zhao, Hanqing Lu, Vincent W. Zheng, Deng Cai, Xiaofei He, and Yueting Zhuang. Community-based question answering via asymmetric multi-faceted ranking network learning. In *Proceedings of the Thirty-First Conference on Artificial Intelligence, February*, pages 3532–3539, San Francisco, California, USA, 2017. AAAI Press.

[Zhou *et al.*, 2012] Tom Chao Zhou, Michael R Lyu, and Irwin King. A classification-based approach to question routing in community question answering. In *Proceedings of the International Conference on World Wide Web*, pages 783–790, Lyon, France, 2012. ACM.