Expertise-Oriented Explainable Question Routing

Yulu Li[1], Wenjun Wang[1,5], Qiyao Peng[2], Hongtao Liu[3], Minglai Shao[2](✉), and Pengfei Jiao[4]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
{liyulu,wjwang}@tju.edu.cn
[2] School of New Media and Communication, Tianjin University, Tianjin, China
{qypeng,shaoml}@tju.edu.cn
[3] Du Xiaoman Financial, Beijing, China
htliu@tju.edu.cn
[4] School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China
pjiao@hdu.edu.cn
[5] College of Information Science and Technology, Shihezi University, Xinjiang, China

Abstract. Question routing aims at routing questions to the most suitable expert with relevant expertise for answering, which is a fundamental issue in Community Question Answering (CQA) websites. Most existing question routing methods usually learn representation of the expert's interest based on his/her historical answered questions, which will be used to match the target question. However, they always ignore the modeling of expert's ability to answer questions, and in fact, precisely modeling both expert answering interest and expertise is crucial to the question routing. In this paper, we design a novel Expertise-oriented Modeling explainable Question Routing (EMQR) model based on a multi-task learning framework. In our approach, we propose to learn expert representation by fully capturing the expert's ability and interest from his/her historical answered questions and the corresponding received vote scores respectively. Furthermore, based on the representations of expert and target question, a multi-task learning model is adopted to predict the most suitable expert and his/her potential vote score, which could provide the intuitive explanation that why routes the question to the expert. Experimental results on six real-world CQA datasets demonstrate the superiority of EMQR, which significantly outperforms existing state-of-the-art methods.

Keywords: Question routing · Community question answering · Recommender systems

1 Introduction

Community Question Answering (CQA) websites have become popular web service which could share and spread knowledge, having drawn much more attention

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 460, pp. 41–57, 2022.
https://doi.org/10.1007/978-3-031-24383-7_3

recently, such as StackOverflow[1] and Quora[2]. A large volume of questions are daily raised on the CQA websites. There are usually low user responses for the majority questions and question raisers need to wait long time to obtain satisfactory answers. Hence, the need for question routing arises in CQA websites. And the question routing aims to recommend suitable experts in the community to answer questions, which could help the question raisers receive satisfactory answers.

Existing approaches for question routing can be categorized into two aspects, including traditional methods and deep learning-based methods respectively. In the traditional approaches, Zhou et al. [34] captures local and global features to explore the possible respondents via a classification task. Liu et al. [18] designs a topic-sensitive probabilistic model to rank user authority, which extracts the topics applying LDA method [1]. However, traditional methods rely on massive labor cost which expend lots of time selecting handcrafted features and are not effective enough to learn complex semantic information. The deep learning-based approaches commonly concentrate on modeling user's expertise by the semantic feature extraction of historical records [8,32].

Despite effectiveness, current deep learning-based methods suffer from the following two key challenges:

- **Expertise Modeling Implicitly.** The great majority of users answering the questions would receive the vote scores from others in the community, which reflect the quality of answers. However, existing deep learning-based methods don't take the score information into account, which could be not comprehensive enough to measure user's expertise. For instance, Fu et al. [8] retrieves similar information between expert historical questions and a raised question to recommend the experts, exploring implicit relevance from the question content.
- **Question Routing Explainablely.** The question routing approaches based on the deep learning usually learn the user representation from his/her historical answered questions to predict the most suitable expert. Although previous methods have achieved great performance, most of them haven't made a reasonable explanation for the routing results. As mentioned above, the answer vote scores could be intuitive explanations whether the users are able to answer questions. Nevertheless, current approaches haven't explored explainable question routing, which would affect the routing credibility.

In order to alleviate the mentioned challenges, we present a novel Expertise-oriented Modeling explainable representation learning approach for the Question Routing (**EMQR**). The core of our approach is to learn expertise-oriented expert representation and utilize the predicted vote scores to enhance the explainability of question routing. To be specific, the question titles and answer scores are encoded into expertise-oriented user representation entangled in the context of semantic and score information of the questions. Moreover, we utilize a

[1] https://www.stackoverflow.com.
[2] https://www.quora.com.

multi-task model to realize the fusion of question routing and recommendation explainability, which not only predicts whether to route new questions to the suitable experts, but also forecasts the answer score to illustrate the explainability of question routing.

The main contributions of our work are summarized as follows:

- We propose an expertise-oriented modeling approach (EMQR) to learn user representation, which is exploited to model expertise explicitly and make explainable question routing effectively. Compared with existing models, EMQR achieves the explainability of routing in addition to question routing.
- We design the multi-task framework that incorporates score encoding into user representation to predict the most suitable expert, which facilitates to learn more appropriate representations for experts. This is a novel attempt to adopt the multi-task learning-based approach to question routing in CQA websites.
- We perform extensive experiments on six real-world datasets from different domains to evaluate the performances of baselines, where EMQR significantly outperforms existing methods over multiple metrics. The experimental results also show that the multi-task framework improves the routing performance.

2 Related Works

In this section, we fall the related works into three aspects, including question routing, multi-task learning and explainable recommendation.

2.1 Question Routing

The question routing aims to recommend suitable experts for the raised questions in CQA websites. And the existing methods can be categorized into traditional methods and deep learning-based methods: **(i) Traditional methods** Traditional methods [2,3,11,15,24,31] have been popular to realize question routing during the earlier periods. Chang et al. [3] proposes a collaborative framework to route the question for a group of users, which introduces the user-user compatibility to build topical expertise via the Linear Regression [27]. Yang et at. [31] designs CQARank model to jointly model textual information and user expertise under different topics. **(ii) Deep Learning-based Methods.** Deep Learning-based methods [9,16,22,23,32] evaluate user expertise via the extraction of semantic information and user features in the neural network. Qiu et al. [23] matches similar questions to route satisfactory experts through a tensor layer, which integrates question representation and semantic retrieval to capture the interaction between different question sentences. Li et al. [16] establishes a heterogeneous information network to incorporate user's profile and utilizes the metapath-based method to learn personalized expert representation in the community. Fu et al. [8] focuses on similar historical information related to new questions and explores the latent relevance between different questions in semantic space.

2.2 Multi-task Learning

The objective of multi-task learning is to improve the generalization by sharing multiple parameters in related tasks [17,28], which has been widely leveraged in recommender systems. And recent approaches [17,19,20,33] primarily introduce deep neural network into the multi-task model. For instance, Ma et al. [20] devises an entire space modeling approach to estimate post-click conversion rate (pCVR) and post-view click-through rate (pCTR) simultaneously for the recommender system. Lin et al. [17] proposes HMTGIN model to concurrently learn the embedding of CQA forum graph on the multi-relational graph, which explicitly models the relationship between question askers and answerers by sharing features. Zhao et al. [33] presents MMoE framework that captures multiple user behavior to optimize engagement objective and satisfaction objective, significantly improving the quality of video watching recommendation.

2.3 Explainable Recommendation

The explainable recommendation attempts to provide the reason why could recommend a particular item to users. And explainable recommender system could improve the acceptance of target items, persuade users to choose the recommended items, and even enhance the overall credibility of system [12]. Many works with respect to the explainable recommendation has been proposed recently [4,5,13,21,30]. For instance, MTER [30] exploits a multi-task learning strategy which integrates two companion tasks via a joint tensor factorization, including user preference modeling for recommendation and opinionated content modeling for explanation. NARRE [4] introduces a novel attention mechanism which could obtain the highly-useful reviews regarded as review-level explanations. RUM [5] introduces memory-augmented neural networks to learn item-level and feature-level representations at a finer granularity, providing reasonable explanations by applying attention mechanism in memory networks.

2.4 Compared with Existing Methods

Although existing methods have achieved great performance, they neglect to model the answer ability of expert for question routing, which would degrade the quality of expert modeling. In this paper, our proposed EMQR aims to model the ability of experts to answer different questions under the influence of score characteristics, which could achieve expertise-oriented explainable representation learning. Furthermore, to enhance the question routing explanation, a multi-task learning framework is employed to perform question routing and provide intuitive explanations for the routing results.

3 Methods

In this section, we first introduce the problem definition for question routing in CQA websites. Next, we present the expertise-oriented explainable question routing approach with multi-task framework (EMQR). The overall architecture is shown in Fig. 1, consisting of three major modules: a *Question Title Encoder* to extract semantic features from the question titles, an *Expertise-oriented Expert Encoder* to capture the expert's interest and expertise with respect to different questions from his/her historical records. Finally, based on the learned expert and target question representations, we employ a *Multi-task Framework* to compute question-expert relevance scores, and simultaneously predict expert potential vote scores for the target question.

3.1 Problem Definition

In this paper, the question routing aims to predict the most suitable expert who can provide an "accepted answer" for the target question in CQA websites. Suppose that the target question is q^t and the candidate expert set is $U = \{u_1, ..., u_m\}$ respectively, where m represents the quantity of candidate experts. For an expert $u_i \in U$, he/she has answered a series of questions denoted as $Q_u^i = \{q_1, ..., q_n\}$, where n represents the size of historical answered question set. And the answers of expert u_i are associated with the vote scores denoted as $V_u^i = \{v_1, ..., v_n\}$. Moreover, the title of j-th answered question q_j is denoted as $W_j = \{w_1, ..., w_l\}$, $j \in [1, n]$, where l is the length of question title. It is notable that the expert who provides the "accepted answer" for target question is regarded as the ground truth.

3.2 Question Title Encoder

In this section, we propose the *Question Title Encoder* to capture the title semantic features of each question. We utilize token embedding layer to project the word sequence of title to a low-dimensional vector sequence. Given a question title q, denoted as $W = \{w_1, ..., w_l\}$, the Byte Pair Encoding [26] is applied to convert the title to token sequence $W^{'} = \{w_1^{'}, ..., w_{l+}^{'}\}$, where $l+$ is the length of token sequence. Then we transform each token in the sequence into a vector, which is denoted as $\mathbf{t} \in \mathcal{R}^{d_w}$ with d_w as dimension. Thus, we can obtain the token embedding matrix $\mathbf{T} = \{\mathbf{t}_1, ..., \mathbf{t}_{l+}\}$, where the dimension of title token sequence is $\mathbf{T} \in \mathcal{R}^{(l+) \times d_w}$. The semantic features of title are extracted through the Convolution Neural Network (CNN) from the token matrix $\mathbf{T}$. Afterwards, we stack the convolution results of K filters denoted as $\mathbf{C}$ and employ a dense layer to convert the dimension from $\mathcal{R}^K$ to $\mathcal{R}^t$:

$$\mathbf{c}_i = \mathrm{ReLU}(\mathbf{W}_\mathrm{i}\mathbf{T} + \mathbf{b}_\mathrm{i}), \mathrm{i} \in [1, ..., \mathrm{K}], \tag{1}$$

$$\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_K], \tag{2}$$

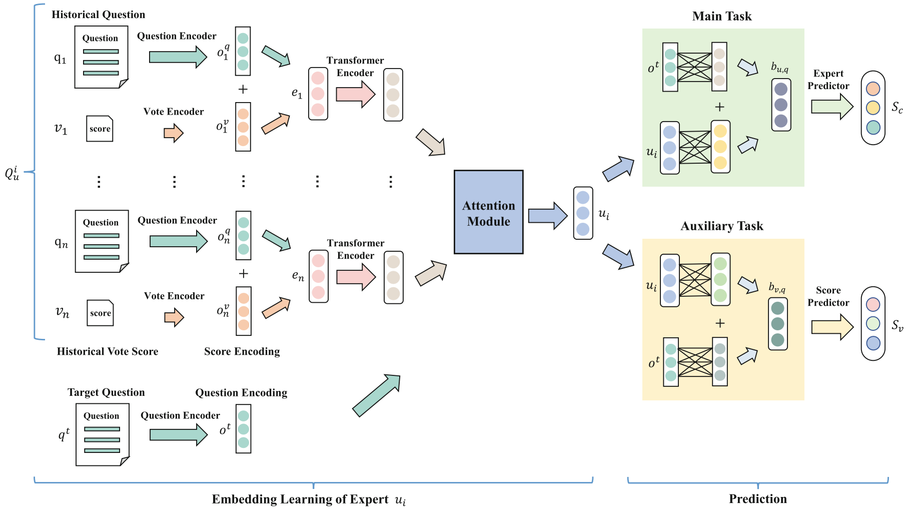


Fig. 1. General architecture of EMQR method.

where $\mathbf{W}_i$ and $\mathbf{b}_i \in \mathcal{R}^{l+}$ are the parameter of i-th filter, $\mathbf{c}_i \in \mathcal{R}^{l+}$ is the i-th convolution result, $\mathbf{C}$ represents the stacked features of a question title and K is the quantity of filters.

In this way, our model could learn the title representation $\mathbf{o} \in \mathcal{R}^t$ via the Question Title Encoder. To distinguish the representations of historical answered questions and target question, we denote the former as $\mathbf{o}_j^q$ and the latter is $\mathbf{o}^t$.

3.3 Expertise-Oriented Expert Encoder

As mentioned above, the expert historical questions and corresponding vote scores could measure his/her interest and expertise in a specific field. The *Expertise-oriented Expert Encoder* aims to model the expert's interest and expertise from his/her historical answered questions. The encoder contains two important modules: (1) a *Transformer Encoder* to unify question title information and vote score features for learning expert's ability to answer different questions; (2) an *Attention Module* to concentrate on more essential and relevant histories for modeling the expert.

Transformer Encoder. For an expert $u_i \in U$, he/she is associated with vote scores which implicitly represent his/her domain ability to answer different questions. Given a vote score $v_j \in V_u^i$ of j-th historical answered question, we encode the score feature and convert it to a low-dimensional vector $\mathbf{o}_j^v \in \mathcal{R}^t$. As shown in Fig. 1, we further combine $\mathbf{o}_j^q$ and $\mathbf{o}_j^v$ of j-th historical question feature by sum operation, denoted as $\mathbf{e}_j$:

$$\mathbf{e}_j = \mathbf{o}_j^q + \mathbf{o}_j^v, j \in [1, n], \tag{3}$$

where $\mathbf{e}_j \in \mathcal{R}^t$ is the representation of j-th historical answered question, which is used as the input of following module.

So as to acquire complex relationship in the long sequence, we take full advantage of the Transformer [29] encoder to learn the aggregated representation of a user's all historical questions, which is utilized to obtain the embedding of j-th historical question feature:

$$\mathbf{e}_j = \mathrm{Trm}(\mathbf{e}_\mathrm{j} + (\mathbf{e}_\mathrm{j})_\mathrm{p}), \tag{4}$$

where $\mathbf{e}_j, (\mathbf{e}_j)_p \in \mathcal{R}^t$ are semantic features and position embedding of j-th historical question, $\mathrm{Trm}(\cdot)$ represents the Transformer encoder as in [29], which includes two sub-layers: a Multi-head Self-attention (MS) layer and a Position-wise fully connected Feed-forward (PF) layer respectively. In order to transmit the information more deeply and enhance the fitting capability of model, the residual connection is utilized in the Transformer layer:

$$\mathrm{Trm}(\mathbf{e}_\mathrm{j}) = \mathrm{LN}(\mathbf{m}_\mathrm{j} + \mathrm{D}(\mathrm{PF}(\mathbf{m}_\mathrm{j}))), \tag{5}$$

$$\mathbf{m}_\mathrm{j} = \mathrm{LN}(\mathbf{e}_\mathrm{j} + \mathrm{D}(\mathrm{MS}(\mathbf{e}_\mathrm{j}))), \tag{6}$$

where $\mathrm{LN}(\cdot)$ is the normalization layer and $\mathrm{D}(\cdot)$ is the dropout layer. Through the MS layer, we acquire expertise features of the user by h attention heads. Afterwards, the multi-head representations are concatenated to obtain the matrix of all historical questions features denoted as $\mathbf{H}_u^i$:

$$\mathbf{H}_u^i = [\mathrm{MS}(\mathbf{e}_1), ..., \mathrm{MS}(\mathbf{e}_\mathrm{n})], \tag{7}$$

$$\mathrm{MS}(\mathbf{e}_\mathrm{j}) = \mathrm{Concat}(\mathbf{h}_1, ..., \mathbf{h}_\mathrm{h})\mathbf{W}_\mathrm{m}, \tag{8}$$

$$\mathbf{h}_i = \mathrm{Att}(\mathbf{e}_\mathrm{j}\mathbf{W}_\mathrm{Q}^\mathrm{i}, \mathbf{e}_\mathrm{j}\mathbf{W}_\mathrm{K}^\mathrm{i}, \mathbf{e}_\mathrm{j}\mathbf{W}_\mathrm{V}^\mathrm{i}), \tag{9}$$

$$\mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\mathrm{T}}{\sqrt{\mathrm{t/h}}}\right)\mathbf{V}, \tag{10}$$

where $\mathbf{H}_u^i \in \mathcal{R}^{t\times n}$ is the feature matrix of expert u_i all historical questions, $\mathrm{MS}(\cdot) \in \mathcal{R}^t$ is the concatenation of representations, $\mathbf{W}_m \in \mathcal{R}^{t\times t}$ is the weight parameter, $\mathbf{W}_Q^i, \mathbf{W}_K^i, \mathbf{W}_V^i \in \mathcal{R}^{t\times t/h}$ are the trainable transformation matrix for the i-th attention head respectively, and $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are both the parameter of attention.

Attention Module. Considering that different historical records have distinct relevances to the target question, we employ the *Attention Module* to pay attention to more essential historical questions under the guidance of target question.

As mentioned above, we have acquired the representation of target question $\mathbf{o}^t$ via the Question Title Encoder. Given the j-th historical feature $\mathbf{h}_j$ in $\mathbf{H}_u^i$, we incorporate target question $\mathbf{o}^t$ into the concatenate feature matrix as follows:

$$\mathbf{a}_u^j = [\mathbf{h}_j \otimes \mathbf{o}^t; \mathbf{h}_j; \mathbf{o}^t], j \in [1, n], \tag{11}$$

where $\mathbf{a}_u^j \in \mathcal{R}^{3\times t}$ is the aggregated representation with respect to concatenate feature matrix and target question, and $\otimes$ represents the element-wise product operation. Afterwards, we adopt the linear layer to project l_j as a vector and calculate the attention weight α_u^j which is computed as:

$$l_j = \mathbf{W}_{j\mathbf{a}_u^j} + b_j, j \in [1, n], \tag{12}$$

$$\alpha_u^j = \frac{\exp(l_j)}{\sum_{i=1}^{n} \exp(l_i)}, \alpha_u^j \in (0, 1), j \in [1, n], \tag{13}$$

where $\mathbf{W}_j$ is the weight parameter, b_j is the bias and α_u^j is the normalized attention weight. Furthermore, we stack all historical questions features and aggregate them with different weights via the sum-pooling operation to obtain final representation $\mathbf{u}_i$ of expert u_i:

$$\mathbf{u}_i = \sum_{i=1}^{n} \alpha^{j_u} \mathbf{h}_j, \tag{14}$$

To sum up, we acquire the final user representation $\mathbf{u}_i$ under the guidance of target question, which will be used for decoding in the following multi-task learning framework to route target question to the most suitable expert.

3.4 Multi-task Learning Framework

In this section, we report the *Multi-task Learning Framework* to decode user representation in order to locate the most suitable expert and acquire expert potential vote score, which includes two types of predictors: the *Expert Predictor* and the *Score Predictor* respectively.

Expert Predictor. For the main task, we first utilize different dense layers to project the expert representation $\mathbf{u}_i$ and target question encoding $\mathbf{o}^t$ separately. And we concatenate them as the input of Expert Predictor. Then we adopt another dense layer to compute the relevance score S_c between the expert and the target question, which is computed as:

$$\mathbf{b}_{u,q} = \mathbf{u}_i \oplus \mathbf{o}^t, S_c = \mathbf{W}_c^T(\mathbf{b}_{u,q}) + \mu_c, \tag{15}$$

where $\oplus$ is the concatenation operation, $\mathbf{b}_{u,q} \in \mathcal{R}^{2\times t}$ is the concatenate representation of expert and target question, $\mathbf{W}_c$ and μ_c are the parameters. Afterwards, the candidate answerer who obtains the highest relevance score will be recommended as the most suitable expert, and the answer provided by him/her will be regarded as the "accepted answer" for target question.

Score Predictor. With respect to the auxiliary task, we conduct potential vote score prediction based on the features of historical records and target question. Analogously, we convert above features to different dense layers and concatenate them as the input of Score Predictor. Then we employ another dense layer to predict the answer score as follows:

$$\mathbf{b}_{v,q} = \mathbf{u}_i \oplus \mathbf{o}^t, S_v = \mathbf{W}_v^T(\mathbf{b}_{v,q}) + \mu_v, \tag{16}$$

where $\mathbf{b}_{v,q} \in \mathcal{R}^{2\times t}$ is the concatenate feature matrix of expert and target question, S_v is the predicted score label, $\mathbf{W}_v$ and μ_v are the parameters. Then the potential vote score provides an intuitive explanation that why recommends the candidate expert to target question. And the higher the potential vote score, the higher the probability which the answer provided by candidate expert will be accepted.

Model Training. In this section, we train our model based on the negative sampling method [14]. And we adopt two loss functions during the train process. Pointing at the main task training, the expert who provides the "accepted answer" is regraded as the positive sample for each question. And other K experts are sampled as negative samples, including answerers who didn't respond this question and who didn't provide acceptable answers. So as to minimize the cross-entropy loss between predicted labels and true labels, the training objective of main task is defined as:

$$\mathcal{L}_{main} = -\sum_{c=1}^{K+1} \hat{S}_c log(S_c), \tag{17}$$

where $\hat{S}_c$ represents the ground truth of user label, and S_c is the probability of expert prediction.

Meanwhile, we also sample N vote score labels for training the auxiliary task. Our purpose in the auxiliary task training is to minimize the cross-entropy loss between predicted score labels and true vote score labels as follows:

$$\mathcal{L}_{aux} = -\sum_{v=1}^{N} \hat{S}_v log(S_v), \tag{18}$$

where $\hat{S}_v$ is the ground truth of vote score label and S_v is the probability for score prediction.

Finally, we unify the expert prediction task with potential vote score prediction task and simultaneously optimize above two tasks with the weight λ, and the total loss of proposed model is defined as follows:

$$\mathcal{L} = \mathcal{L}_{main} + \lambda \mathcal{L}_{aux}. \tag{19}$$

3.5 Differences with Existing Techniques

In this section, we highlight the key differences between the proposed method and existing techniques.

RMRN. RMRN [8], a recent deep learning-based method for the question routing, exploits recurrent memory reasoning network to explore the relevance between user historical records and target question. Both our proposal and RMRN learn expert representations from his/her historical answered questions, which recommend suitable experts for raised questions. For the expertise modeling, RMRN couldn't explicitly measure expert's ability to answer questions. Unlike RMRN, our approach leverages expertise-oriented expert encoder to model expert's interest and expertise, instead of a complex recurrent structure equipped with cascading reasoning memory cells, control and memory units.

NeRank. NeRank [16], a recently proposed method to achieve personalized question routing, utilizes heterogeneous information network embedding algorithm to learn user representation which takes question raiser's profile into account. Compared with NeRank, EMQR provides intuitive explanations for routing results via the multi-task learning framework. Thus, our model not only recommends experts precisely when new questions are raised, but also makes the explanations for recommended experts improving user credibility for CQA websites. Moreover, our method adopts a score prediction module to forecast the answer score corresponding to target question.

4 Experiments

In this section, we further conduct experiments on six real-world datasets to evaluate the performance of proposed method, EMQR. First, we introduce the datasets and experiment settings successively. Then the performance is evaluated against several existing methods in terms of three metrics. And we dissect the impact of vote score features on the model performance with an ablation study. In addition, the parameter sensitivity experiment is performed to explore the effect of auxiliary hyper-parameter. Finally, in order to make it suitable for the question routing, we demonstrate the explainability of our proposed EMQR via a case study.

4.1 Datasets and Experimental Settings

Datasets. Six real-world datasets related to the CQA are applied to conduct the experiments, including six domains, i.e., Print, History, Bioinformatics, AI, Biology and English. Each dataset contains a question set where each question includes its title and a vote score, which indicates how satisfied other users in the community are with the answer. Following the work [16], we reserve the experts who have answered at least 5 questions to avoid cold start problem. Table 1 shows the statistics of six datasets in details.

In chronological order, each dataset is divided into a training set (80%), a validation set (10%) and a testing set (10%) respectively. For each question, a candidate expert set is built with 20 experts which includes respondents whose answers were not accepted and others who didn't answer the question. In addition, the expert who provides the "accepted answer" for target question is the ground truth.

Table 1. Statistics of the datasets.

Dataset	#question	#answerer	#answer	#avg.title length	#avg.vote	#vote range
AI	1,205	195	1,719	1097	7.20	−3∼136
Print	1,033	112	1,686	9.51	9.24	−6∼44
Biology	8,704	630	11,411	9.84	9.59	−7∼241
English	46,692	4,781	104,453	9.68	16.34	−69∼828
History	4,904	471	9,452	12.38	29.57	−14∼292
Bioinformatics	958	113	1,489	9.93	7.72	−5∼142

Hyper-parameter Setting. In our experiment, the dimensions of token embedding and question features are set to 100. And the quantity of Transformer heads and Transformer encoder layers are both 2 respectively. We employ truncate or pad operations to fix the length of historical answered question to 30. For each question, the length of question title is fixed to 15. In order to balance two tasks in our model, we set the auxiliary weight λ as 0.9.

Evaluation Metrics. We adopt common recommendation ranking metrics to evaluate our approach following the literature [32], which include MRR [7], Precision@K where $K = 1$ and $K = 3$.

Baselines. We compare the proposed EMQR with eight competitive methods: **(1) Score:** A feature engineering-based approach selects the answerer who has greater quantities of "accepted answer" as the recommended expert. **(2) BM25** [25]**:** A text-retrieval algorithm calculates the short-text similarity between historical document collection and a given document. **(3) Doc2Vec:** A document representation method learns semantic encoding from historical answered questions to obtain the relevance in regard to target question, which predicts a vector to represent different documents. **(4) CNTN** [23]**:** The CNTN approach extracts semantic information of the question by integrating question representation and word token matching. **(5) RMRN** [8]**:** The RMRN method adopts a recurrent memory reasoning network to retrieve historical information as to the target question, exploring implicit relevance of textual content. **(6) TCQR** [32]**:** The model considers temporal characteristics to acquire the expert's expertise dynamically under the multi-shift and multi-resolution. **(7) NeRank** [16]**:** NeRank constructs heterogeneous information network to learn three representations of questions, raisers and answerers, and designs a convolutional scoring function to calculate the ranking score. **(8) UserEmb** [9]**:** The approach explores text and node similarity to model expert's interest in social network, which applies word2vec [6] to obtain semantic encoding of the question and node2vec [10] for learning the node representation.

4.2 Performance Comparison

Table 2 reports the results compared our model EMQR with above baselines on six real-world datasets. We observe that the proposed model (EMQR) achieves more excellent performance than all baselines, which demonstrates that our approach can effectively recommend the most suitable expert for target question.

MRR metric in each dataset indicates a huge improvement for all baselines in terms of the overall ranking performance.

Specifically, our model achieves 37.16% P@1 on History dataset, meaning that around 37.16% of the results where correct candidate experts have 37.16% probability to be ranked first, which is significantly better than 5.26% P@1 of the best baseline method, RMRN.

In addition, EMQR outperforms the best baseline over P@3 metric on all datasets, which demonstrates our proposed method can effectively recommend the expert who provides the "accepted answer" when new question is raised.

From the comparison results of all baselines, we can also conclude as follows: 1) We observe that the models (RMRN, NeRank and EMQR) which capture the interaction under the guidance of target question consistently outperform than those not (Doc2Vec, CNTN, TCQR and UserEmb). The reason is the methods pay attention to capturing expert's interest from historical records which are more similar to target question under the finer granularity. 2) Our proposal achieves obvious improvement compared to all baselines, since we integrate vote score features into user expertise to achieve expert representation learning for distinct expertise. EMQR describes expert's ability to answer the questions from different domains, while other methods don't take vote score features into account for question routing. For instance, RMRN discovers the latent relevance of semantic information related to a new question over the historical records, and NeRank recommends the expert which regards user ID as a personalized characteristic in heterogeneous network. 3) We demonstrates that our proposed EMQR based on multi-task framework improves the explainability of recommended experts, which routes the questions to answerers with higher vote scores in related domains.

4.3 Ablation Study

Since the vote scores play a critical role on modeling the user expertise, we further study the effectiveness of score features in this section. We conduct an ablation experiment in the absence of vote score features denoted as *w/o Score*.

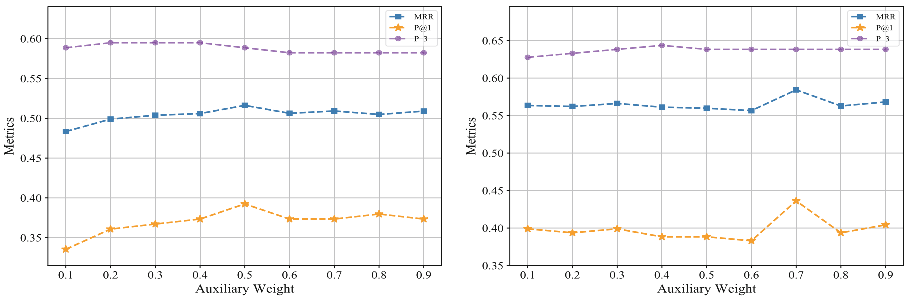
As shown as in Table 3, removing score information obviously reduces the model performance across three metrics, due to w/o Score variant is insufficient to represent user's expertise for question routing. The results indicate that our modeling choice of score features for each question is appropriate to tackle the expertise integration challenge involved in question routing task. In addition, we can conclude that the vote score features have significant impact on modeling expert representation.

Table 2. Performances (%) of the baselines on six datasets.

Dataset	Print			History			Bioinformatics		
Method	Metric								
	MRR	P@1	P@3	MRR	P@1	P@3	MRR	P@1	P@3
Score	19.32	10.67	19.36	18.97	6.411	19.76	17.89	10.23	23.39
BM25	35.68	18.72	40.11	25.91	15.41	26.74	28.73	15.16	39.27
Doc2Vec	39.77	19.62	39.22	28.43	17.43	29.87	32.64	18.13	44.23
CNTN	47.32	29.11	58.05	39.31	25.34	43.05	43.37	25.31	50.62
RMRN	48.44	29.79	<u>59.57</u>	<u>52.21</u>	<u>33.51</u>	<u>68.86</u>	44.44	26.58	53.16
TCQR	44.25	26.67	51.11	40.21	27.37	47.02	39.46	26.01	43.18
NeRank	<u>51.33</u>	<u>31.65</u>	59.37	46.75	27.73	56.67	<u>45.01</u>	<u>28.11</u>	<u>54.68</u>
UserEmb	36.39	28.34	48.31	40.39	26.39	47.42	37.89	25.05	43.03
EMQR	**52.96**	**35.17**	**60.09**	**55.59**	**37.16**	**69.11**	**47.96**	**33.28**	**58.23**
Dataset	AI			Biology			English		
Method	Metric								
	MRR	P@1	P@3	MRR	P@1	P@3	MRR	P@1	P@3
Score	22.32	9.851	24.29	19.73	10.17	13.76	17.21	7.761	19.74
BM25	35.64	15.47	40.15	27.12	14.86	25.76	20.13	14.21	28.77
Doc2Vec	38.21	17.78	43.32	29.12	16.08	28.73	23.26	15.23	29.46
CNTN	45.06	27.78	54.44	33.58	20.17	33.69	29.68	18.37	36.39
RMRN	45.24	27.83	<u>62.22</u>	<u>43.62</u>	<u>24.53</u>	<u>55.62</u>	46.77	25.22	<u>61.62</u>
TCQR	41.96	27.78	45.59	39.62	24.06	44.22	34.25	19.27	49.87
NeRank	<u>49.89</u>	<u>33.03</u>	62.04	41.71	23.86	47.61	<u>48.95</u>	<u>27.16</u>	61.43
UserEmb	41.01	23.35	44.57	32.23	19.87	32.75	31.73	19.56	42.36
EMQR	**50.96**	**34.44**	**64.38**	**48.19**	**30.31**	**58.44**	**51.35**	**31.85**	**62.73**

Table 3. Effectiveness of vote score features.

Dataset	Print			Biology			Bioinformatics		
Method	Metric								
	MRR	P@1	P@3	MRR	P@1	P@3	MRR	P@1	P@3
w/o Score	37.99	25.83	52.13	45.16	27.49	54.85	40.15	31.52	53.80
EMQR	**52.96**	**35.17**	**60.09**	**48.19**	**30.31**	**58.44**	**47.96**	**33.28**	**58.23**



(a) Bioinformatics (b) Print

Fig. 2. The impact of hyper-parameter.

4.4 Parameter Sensitivity Analysis

To investigate the impact of auxiliary task on the performance of EMQR, in this section, we conduct parameter sensitivity analysis for the hyper-parameter λ. Figure 2 shows the results of hyper-parameter experiment with the auxiliary weight λ ranging from 0.1 to 0.9 in Bioinformatics and Print datasets. As shown in Fig. 2, we observe that the performance continues to improve with the increase of auxiliary weight. If the auxiliary weight exceeds a turning point, the performance begins to decrease gradually due to the over fitting. For example, the performance of Bioinformatics dataset starts to decline when the auxiliary weight is greater than 0.5. And the results in Print dataset indicate EMQR performance achieves the best when the auxiliary weight is set to 0.7.

4.5 Case Study

In order to verify that EMQR can enhance the explainability of question routing, we show a case study about experts receiving scores for the "accepted answer". We randomly select several questions over three datasets (AI, Bioinformatics and Biology), showing the title of question (i.e., Question), the answerer of question (i.e., Answerer), the potential vote score predicted by Score Predictor (i.e., Score) and the reception of answer (i.e., Reception) in Table 4. Each question receives numerous answers provided by the experts from different fields, however, only one candidate expert who acquires the higher score is accepted as the most suitable expert. From the results, we can observe that the experts obtaining higher scores are respectively recommended to target questions. For instance, the question "why is turner syndrome a problem?" in Biology dataset is separately answered by the Answerer c and the Answerer d. And the answer provided by the former gains 6 points with a higher score, while the answer of the latter receives 4 points with a lower score. Hence, our proposed EMQR not only precisely routes questions to the experts, but also makes the routing results explainable in CQA websites.

Table 4. The case study of vote scores with respect to the "accepted answer" in different datasets.

Dataset	Question	Answerer	Score	Reception
AI	is fuzzy logic invalid?	a	4	Accepted
		b	1	No
Biology	why is turner syndrome a problem?	c	6	Accepted
		d	4	No
Bioinformatics	cds length for each human gene?	e	4	Accepted
		f	0	No
	what does pca mean on gwas?	g	3	Accepted
		h	2	dejkfvcNo

5 Conclusion

In this paper, we propose an expertise-oriented explainable question routing model with a multi-task framework, EMQR, which can precisely recommend suitable experts for questions in CQA websites. Our approach learns the expert representation by combining semantic information and score characteristics, which reflects the expert's interest and expertise for answering different questions. Afterwards, we leverage the multi-task framework to realize expert prediction and potential vote score prediction, which could route target question to the most suitable expert and provide the reason why routing. Compared to existing state-of-art methods, the proposed EMQR achieves more remarkable performance in extensive experiments and enhances the explainability of question routing.

Acknowledgments. This work was supported by the China Postdoctoral Science Foundation (2022T150470, 2021M702448), the Sustainable Development Project of Shenzhen (KCXFZ20201221173013036).

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Cao, X., Cong, G., Cui, B., Jensen, C.S., Yuan, Q.: Approaches to exploring category information for question retrieval in community question-answer archives. ACM Trans. Inf. Syst. (TOIS) **30**(2), 1–38 (2012)
3. Chang, S., Pal, A.: Routing questions for collaborative answering in community question answering. In: Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, pp. 494–501. IEEE (2013)
4. Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: Proceedings of the 2018 World Wide Web Conference, pp. 1583–1592 (2018)

5. Chen, X., et al.: Sequential recommendation with user memory networks. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 108–116 (2018)
6. Church, K.W.: Word2vec. Nat. Lang. Eng. **23**(1), 155–162 (2017)
7. Craswell, N.: Mean reciprocal rank. Encyclopedia of Database Systems, vol. 1703 (2009)
8. Fu, J., et al.: Recurrent memory reasoning network for expert finding in community question answering. In: WSDM, pp. 187–195 (2020)
9. Ghasemi, N., Fatourechi, R., Momtazi, S.: User embedding for expert finding in community question answering. ACM Trans. Knowl. Discov. Data **15**(4), 1–16 (2021)
10. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
11. Guo, J., Xu, S., Bao, S., Yu, Y.: Tapping on the potential of Q&A community by recommending answer providers. In: Proceedings of the ACM Conference on Information and Knowledge Management, pp. 921–930 (2008)
12. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, pp. 241–250 (2000)
13. Hou, Y., Yang, N., Wu, Y., Yu, P.S.: Explainable recommendation with fusion of aspect information. World Wide Web **22**(1), 221–240 (2019)
14. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using click through data. In: Proceedings of the Conference on Information and Knowledge Management, pp. 2333–2338 (2013)
15. Ji, Z., Wang, B.: Learning to rank for question routing in community question answering. In: Proceedings of the ACM International Conference on Information & Knowledge Management, pp. 2363–2368 (2013)
16. Li, Z., Jiang, J.Y., Sun, Y., Wang, W.: Personalized question routing via heterogeneous network embedding. In: Proceedings of the International Conference on Artificial Intelligence, vol. 33, pp. 192–199 (2019)
17. Lin, Z., et al.: Multi-relational graph based heterogeneous multi-task learning in community question answering. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 1038–1047 (2021)
18. Liu, X., Ye, S., Li, X., Luo, Y., Rao, Y.: ZhihuRank: a topic-sensitive expert finding algorithm in community question answering websites. In: Li, F.W.B., Klamma, R., Laanpere, M., Zhang, J., Manjón, B.F., Lau, R.W.H. (eds.) ICWL 2015. LNCS, vol. 9412, pp. 165–173. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25515-6_15
19. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1930–1939 (2018)
20. Ma, X., et al.: Entire space multi-task model: an effective approach for estimating post-click conversion rate. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 1137–1140 (2018)
21. Peake, G., Wang, J.: Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2060–2069 (2018)

22. Qian, Y., Tang, J., Wu, K.: Weakly learning to match experts in online community (2016)
23. Qiu, X., Huang, X.: Convolutional neural tensor network architecture for community-based question answering. In: Proceedings of the International Joint Conference on Artificial Intelligence (2015)
24. Riahi, F., Zolaktaf, Z., Shafiei, M., Milios, E.: Finding expert users in community question answering. In: Proceedings of the International Conference on World Wide Web, pp. 791–798 (2012)
25. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009)
26. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the International Conference on Association for Computational Linguistics (2016)
27. Su, X., Yan, X., Tsai, C.L.: Linear regression. Wiley Interdiscip. Rev. Comput. Stat. **4**(3), 275–294 (2012)
28. Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L.: Multi-task learning for dense prediction tasks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. (2021)
29. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the International Conference of Neural Information Processing Systems, pp. 5998–6008 (2017)
30. Wang, N., Wang, H., Jia, Y., Yin, Y.: Explainable recommendation via multi-task learning in opinionated text data. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 165–174 (2018)
31. Yang, L., et al.: CQARank: jointly model topics and expertise in community question answering. In: Proceedings of the ACM International Conference on Information & Knowledge Management, pp. 99–108 (2013)
32. Zhang, X., et al.: Temporal context-aware representation learning for question routing. In: WSDM, pp. 753–761 (2020)
33. Zhao, Z., et al.: Recommending what video to watch next: a multitask ranking system. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 43–51 (2019)
34. Zhou, T.C., Lyu, M.R., King, I.: A classification-based approach to question routing in community question answering. In: Proceedings of the International Conference on World Wide Web, pp. 783–790 (2012)