Computers, Materials & Continua
DOI: 10.32604/cmc.2023.035239
Article

Tech Science Press

Check for updates

Expert Recommendation in Community Question Answering via Heterogeneous Content Network Embedding

Hong Li[1,*], Jianjun Li[1], Guohui Li[1], Rong Gao[2] and Lingyu Yan[2]

[1]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[2]College of Computer Science and Technology, Hubei University of Technology, Wuhan 430068, China
*Corresponding Author: Hong Li. Email: li0405@hust.edu.cn
Received: 13 August 2022; Accepted: 08 December 2022

Abstract: Expert Recommendation (ER) aims to identify domain experts with high expertise and willingness to provide answers to questions in Community Question Answering (CQA) web services. How to model questions and users in the heterogeneous content network is critical to this task. Most traditional methods focus on modeling questions and users based on the textual content left in the community while ignoring the structural properties of heterogeneous CQA networks and always suffering from textual data sparsity issues. Recent approaches take advantage of structural proximities between nodes and attempt to fuse the textual content of nodes for modeling. However, they often fail to distinguish the nodes' personalized preferences and only consider the textual content of a part of the nodes in network embedding learning, while ignoring the semantic relevance of nodes. In this paper, we propose a novel framework that jointly considers the structural proximity relations and textual semantic relevance to model users and questions more comprehensively. Specifically, we learn topology-based embeddings through a hierarchical attentive network learning strategy, in which the proximity information and the personalized preference of nodes are encoded and preserved. Meanwhile, we utilize the node's textual content and the text correlation between adjacent nodes to build the content-based embedding through a meta-context-aware skip-gram model. In addition, the user's relative answer quality is incorporated to promote the ranking performance. Experimental results show that our proposed framework consistently and significantly outperforms the state-of-the-art baselines on three real-world datasets by taking the deep semantic understanding and structural feature learning together. The performance of the proposed work is analyzed in terms of MRR, P@K, and MAP and is proven to be more advanced than the existing methodologies.

Keywords: Heterogeneous network learning; expert recommendation; semantic representation; community question answering

1 Introduction

Recent years have witnessed a spectacular increase in real-world applications of Community Question Answering (CQA), such as Quora, Stack Overflow, and Zhi Hu. On these online service platforms, people exchange knowledge and information in the form of posting questions and providing answers almost every second. According to statistics, as of November 2020, the Stack Overflow community has accumulated more than 14 million registered users, who have asked 21 million questions and received 31 million answers [1]. Unfortunately, not all questions have been answered in time, and even a large number of questions have been put on hold for a long time without getting attention. Meanwhile, it is difficult for a potential expert with professional knowledge and answering willingness to find suitable questions to provide answers. The gap between the pending questions and the potential experts has seriously hindered knowledge sharing and experience exchange in CQA. Therefore, the expert recommendation task aims to identify domain expert users and push unresolved questions to appropriate experts to obtain fast and high-quality answers is an effective strategy to bridge this gap. Most traditional methods are textual content-based, which construct user profiles by collecting their activity text evidence stored in CQA achieves and take the expert recommendation as a text matching task. Specifically, they turn the question sequence and the user profile into dense meaningful feature vectors by using language models [2,3], topic models [4–6], or deep neural networks (e.g., CNN's or LSTM's) [7,8], and then conduct further matching based on learned embeddings. Although these methods have achieved promising performance [6,9,10], especially with the significant improvement brought by deep learning methods [7,11,12], several challenges have not been addressed so far. First, content-based methods have been plagued by data sparseness issues for a long time since the average participation rate of users in the Q&A community is low. Most of the answers come from a small number of users, resulting in very sparse text information available to evaluate users [13,14]. Take the user's activities in Quora as an example [15,16], 90% of questions on Quora have less than ten answers and Over 30% of Quora registered users haven't responded to any question, the number of users who answered more than 4 questions is only 16.74%. Second, the capacity of text understanding of currently used deep neural networks (e.g., CNN's or LSTM's) is still limited, especially when dealing with short questions. Third, content-based methods focus on textual information analysis, while ignoring the structural features of the CQA network and the relationships between different entities. A CQA network is a typical heterogeneous information network (HIN) in which the heterogeneous information provided by different types of entities and edges retains more semantically meaningful information than homogeneous networks. Taking the Stack Overflow platform as an example, as illustrated in Fig. 1a, which consists of three entity types: Question, Answerer, and Tag, and two relationships: an answerer provides an answer to a question, and a tag belongs to a question to indicate the knowledge domain of a question. We can introduce two meta-paths: "Answerer-Question-Answerer (AQA)", and "Answerer-Question-Tag-Question-Answerer (AQTQA)". The semantics of the two meta-paths give two specific definitions of how the two answerers are seen as relevant or similar. Therefore, intuitively, we can conclude that making use of the structure and relationship information hidden in the CQA network will greatly improve the performance of the expert finding and recommendations. Moreover, a lot of work have shown that encoding the interaction activities between nodes is invaluable for node modeling in CQA [17,18]. Compared to content-based methods, a few approaches based on network topology learning have emerged to address the expert recommendation task in the community more recently. Especially the random-walk-based methods [7,18–20] and meta path-based methods [21]. They focused on analyzing structural proximity between nodes and attempted to fuse the textual content of nodes during the learning process. However, they usually only consider the content of a part of the nodes (usually the nodes whose node type is in question) and ignore the semantic correlation between different types of nodes, resulting in the loss of semantic information. Furthermore, in random walk-based methods, the transitions between nodes are selected completely randomly, lacking the consideration of heterogeneous information of network nodes, such as the type of nodes. Meanwhile, in existing meta path-based methods, meta-paths are not distinguished and they assume the same weights of meta paths for all nodes, which results in the inability to capture the personalized preferences of each node on meta paths. Still taking the Stack Overflow network as an example, sometimes it can be considered that different users who answer the same question are close, and sometimes even though two different users answer different questions, if these questions involve the same domain, the two different users may also be considered to be close to each other in a specific field and have a strong interest in the same topic. Furthermore, given a certain meta path, i.e., Answerer-Question-Tag-Question-Answerer, an answerer can be connected to other answerers by different path instances. However, the tags connecting them may be rough or coincidental, and there may be several tags connected to the question. Therefore, distinguishing different meta paths and path instances in the CQA network embedding methods can highlight the most relevant information and ignore the noise to obtain better node embeddings.

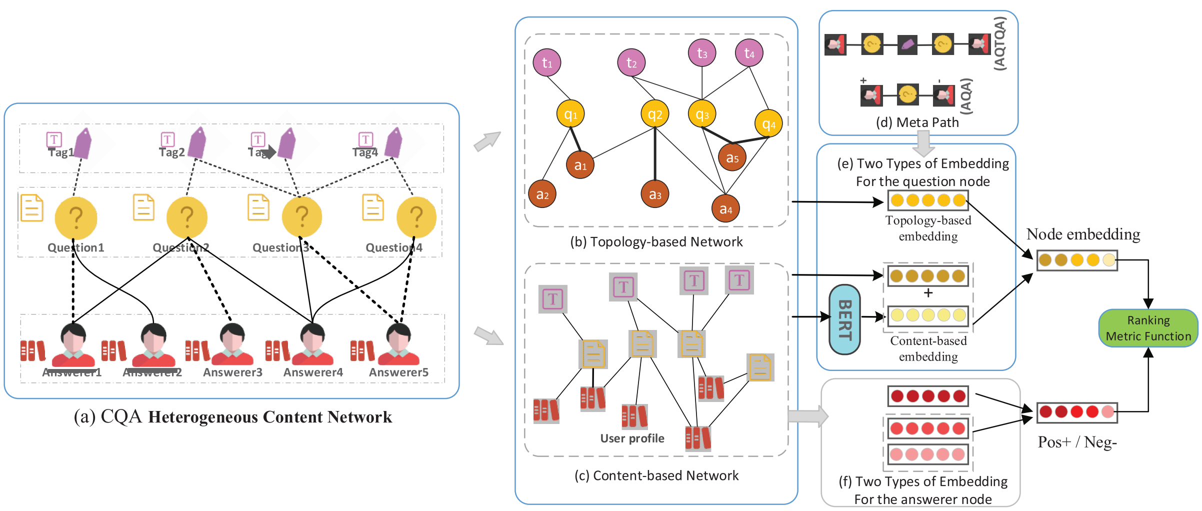


Figure 1: The overview of combined node embedding learning for CQA heterogeneous content network (a) The CQA heterogeneous content network is constructed by integrating structural proximity as well as node content information. (b) The extracted topology-based network. (c) The extracted content-based network. (d) The meta path schemes. (e) Two types of embedding for the question node. (f) Two types of embedding for the answerer node

To solve the above limitations, in this paper, we propose a novel model ER-HCNE, which stands for **E**xpert **R**ecommendation via **H**eterogeneous **C**ontent **N**etwork **E**mbedding. ER-HCNE considers both network structural information and textual content information to find suitable domain experts to answer a given question in a CQA heterogeneous network. Specifically, we construct a competitive heterogeneous content network according to users' behavior in the CQA and develop it into a topology-based network and a content-based network to learn node representations from two different perspectives. A hierarchical attentive network learning algorithm is deployed to learn the topology-based embedding, aiming to preserve nodes' structural proximity information and personalized preferences. The content-based embedding is constructed by combining the textual correlations between adjacent nodes learned by a meta-context-aware skip-gram model and the textual representation of nodes learned from a pre-trained deep bidirectional representation model. In addition, the rich non-linear ranking information contained between answers is also used to calculate the final relative quality ranking score of the answerers.

The rest of this paper is organized as follows. In Section 2, a brief review of the most relevant work is mentioned. Then, we gave the problem formulation in Section 3. In Section 4, a novel framework is introduced and its details are described. And in Section 5, the experiment settings and related results are presented and compared with other methods. In addition, we conducted in-depth discussions on several aspects. Finally, Section 6 gives conclusions and future work.

2 Related Works

2.1 Expert Recommendation

Two research works that have been extensively studied in the social community are very close to our expert recommendation tasks, namely expert finding and question routing. In most cases, these three tasks can be considered equivalent. Methods to solve them can be broadly divided into two categories: text representation-based and network embedding-based.

The text representation-based approaches mainly focus on word representation and text semantic understanding. It usually represents the users with textual profiles according to their answering record left in CQA achieves and take the problem as a text matching task, aiming to find out the most relevant users' profile to the newly posted question content. Traditional models includes statistical language models [9] and topic models [22–24] rely heavily on term overlap or word co-occurrent, resulting in limited text understanding ability to distinguish deep semantic features. In recent years, deep learning neural networks is proposed to address the deep semantic patterns extraction problem, leading to a significant performance improvement. For example, Azzam et al. [25] directly applied a deep neural network DSSM (Deep Semantic Similarity Model) [26] to map the question and user's profile to a low-dimensional semantic space for more meaningful representation. It can be said that methods based on neural networks (i.e., CNNs, RNNs, etc.) for expert recommendation tasks have been more popular and effective for a long time. However, the limited receptive field to capture long-distance dependencies lies in CNNs and the difficulty to parallelize lies in RNNs severely drags down the performance of these approaches. In addition, text representation-based methods mainly focus on deep textual content understanding, while ignoring the network structural proximity and relations between different entities. Furthermore, many recent works [27,28] point out that the rising star Transformer [29] and BERT [30] far outperform CNNs and RNNs on text semantics learning and sequence relation exploring.

The network embedding-based approaches address the problem from another perspective by focusing on network structure analysis. Random-walk-based methods [31–33], which adopt a random walk strategy to expand the neighborhood of a vertex in large-scale networks to learn latent representations of nodes, obtain a lot of attention in recent years. Chen et al. [34] both adopted Deep Walk [31] to exploit the plentiful social information to solve the data sparsity problem. However, the main limitation of these random walk-based methods is that they are suitable for learning relations in homogeneous networks, but not sufficient for relation exploration in the multi-type and complex heterogeneous social networks we need to deal with. The same author proposed another model named APT [35] to tackle the challenge of directed graph embedding, which only considers user interactions in the community without incorporating textual content.

To overcome the shortcomings of the above methods and build more comprehensive node representations for expert recommendation tasks, in our work, we use the textual content of various types of nodes to learn content-based embedding and explore the structural properties between different nodes through a hierarchical attentive network learning model, while retaining the individualized preferences of nodes.

2.2 Heterogeneous Information Network Embedding

In recent years, the development of network embedding technology has highly promoted the analysis and understanding of social networks. At the same time, inspired by deep learning and Word2vec [36], several effective network embedding models such as DeepWalk [31], LINE [33], and Node2vec [32] have emerged and is widely used in different tasks. However, these methods are designed for learning node representations in homogeneous networks whereas CQA networks are heterogeneous that consist of varieties of entities and complex links. Compared to homogeneous networks, the goal of heterogeneous network embedding focus on retaining the structure and relational properties of the network while projecting the nodes into the potential embedding space [37]. The learned node vector can be used as the input of deep learning related algorithms to complete tasks such as question answering, node classification, and social network analysis.

Several attempts have been made in HIN embedding and achieved promising performance in various domains. Tang et al. [38] designed a skip-gram based model called PTE decomposes the input heterogeneous network into several homogeneous networks and then performs network embedding individually. Dong et al. [39] proposed Metapath2vec in 2017, which uses meta path based random walk to construct the heterogeneous neighborhood of each node. In this method, the walker is restricted to transition between different types of nodes in a unified way instead of the random walk. Compared with PTE, Metapath2vec can better capture the structural dependencies between different types of nodes. Later, various approaches leverage meta paths to construct heterogeneous contexts for learning embeddings. Fu et al. [40] proposed an approach that directly uses meta paths as contexts to learn embeddings for vertices by jointly conducting multiple prediction training tasks. HeteSpaceyWalk [41] exploits the heterogeneous personalized spacey random walk to learn embeddings for multiple types of nodes guided by meta paths, graphs, and schemas, respectively. Although the above-mentioned-methods have utilized meta paths to know the comprehensive proximity and semantics between nodes, they do not distinguish meta paths or path instances for nodes.

The major drawbacks of the existing methodologies is the inefficiency in determining and distinguishing the meta paths between the nodes or the instance path of nodes. The setback in determining the meta path, path based context are the major challenging factors of the existing methods.

3 Problem Formulation

In this section, we introduce the concepts of heterogeneous content network, meta path, meta path instance, meta path-based context, and node embedding. Finally, we formally define the expert recommendation problem.

Definition 1 (heterogeneous content network). A heterogeneous content network (HCN) is defined as a network $G = (V, E, C)$, where V is the set of nodes, $E = \{e_{ij}\}_{i,j=1}^{|V|}$ is the set of edges, and $C = \{c_1, c_2, \ldots, c_{|V|}\}$ denotes the text information of nodes. Specifically, e_{ij} denotes the relationship between two nodes (v_i, v_j) linked with each other, c_i is a word sequence associated with node v_i denoted as

$c_i = \{w_1, w_2, \ldots, w_{|c_i|}\}$, where $|c_i|$ is the number of words in c_i. $\phi: V \to T_V$ and $\psi: E \to T_E$ are two types of mapping functions associated with each node and edge, respectively. Each node $v_i \in V$ is mapped to one particular node type in T_V, and each edge $e \in E$ is mapped to one particular edge type in T_E, where $|T_V| + |T_E| > 2$.

In our task, the dataset is built upon the static archive of the CQA website, which keeps all the Q&A records accumulated over time. We create the Questions set $Q = \{q_1, q_2, \ldots, q_k\}$ and Answerers set $A = \{a_1, a_2, \ldots, a_m\}$, separately, where k is the number of questions and m is the number of users who have answered questions. Meanwhile, each question has several different tags to identify the knowledge domain to which it belongs. We define the Tags set $T = \{t_1, t_2, \ldots, t_i\}$, where i is the number of tags. The CQA network in big Q&A data can be seen as an HCN, as illustrated in Fig. 1a, in which $V = Q \cup A \cup T$. A question could be answered by different users, and a tag could belong to different questions. It is worth noting that there is only one best answerer for a certain question (the black bold dotted line means "provides the best answer" in Fig. 1a).

We emphasize that the CQA heterogeneous content network defined in our model is different from the commonly defined HIN network since we consider the content related to each node instead of only focusing on structural proximity between nodes.

Definition 2 (meta path). A meta path λ is defined as a sequence of relations connecting different types of nodes in the form of $p_1 \xrightarrow{l_1} p_2 \xrightarrow{l_2} \ldots p_t \xrightarrow{l_t} p_{t+1} \ldots \xrightarrow{l_{k-1}} p_k$. $l = l_1 \circ l_2 \circ \ldots \circ l_{k-1}$ denotes the composite relations between node types p_1 to p_k, $p_i \in T_V$.

For example, in Fig. 1d, the meta-path "AQA" extracted from the network denotes the co-answer relationship of a question between two answerers. In addition, the two answerers can also be connected by another meta-path "AQTQA", which shows that even if the two answerers do not appear in the same question thread, we can also infer from the links that they have similar expertise backgrounds and interests, consistent with the overlapping knowledge areas covered by the questions.

Definition 3 (meta path instance). Given a meta path λ of a heterogeneous graph, a meta path instance I_λ is defined as a node sequence in the network which generates according to the sequence of types in λ. When a meta instance connects node v_i and node v_j where the node v_i is the target node, we denote this meta instance as $I_{\lambda i}$.

Definition 4 (meta path based context, MPBC). Given a meta path instance $I_{\lambda i}$, the meta path λ based context of the target node v_i is denoted as $\mathbb{C}_i^\lambda$, which is defined as the node sequence in the rest of the meta path instances excluding v_i.

Assume $I_\lambda = \{I_{\lambda 1}, \ldots, I_{\lambda N}\}$ is the set of generated meta path instance. $\mathbb{C}_i$ is the context of v_i. **If** $I_{\lambda 1} = v_1 v_2 v_3 v_4 v_5 v_6 v_7 \ldots$. and the context window size is 2, then $\{v_2, v_3, v_5, v_6\} \in \mathbb{C}_4$ denotes the context of v_4. In addition, we regard the node pair $\{(v_4, v_2), (v_4, v_3), (v_4, v_5), (v_4, v_6)\}$ as the positive samples of node v_4.

Definition 5 (node embedding). Given a heterogeneous content network $G = (V, E, C)$, the node embedding aims to project each node $v_i \in V$ into a low-dimensional latent space where each node can be represented as $e \in R^d (d \ll |V|)$. The learned representation for each node e is supposed to preserve both the network structural proximity information and textual relevance information.

Definition 6 (expert recommendation problem) The overall view of our expert recommendation problem is illustrated in Fig. 2. Given a set of archived question sessions including answerers list, tags list, and content information (i.e., question's title and body, answerers' profile, and content of tags), our task is to train a model and rank all potential answerers for a newly posted question, the top-ranked answerer is recommended as the domain expert for the question. In summary, we declare the main Notations in Table 1.

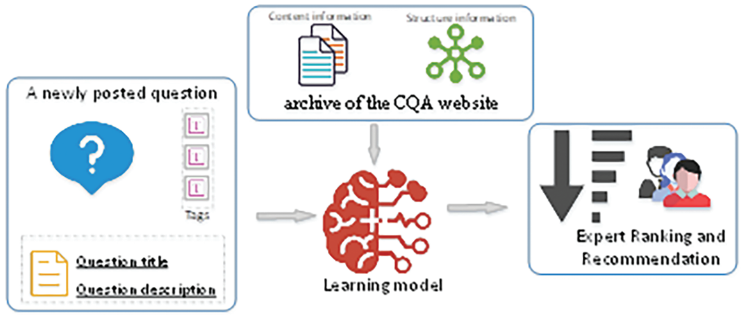


Figure 2: Overall view of expert recommendation problem

Table 1: Notations

Symbol	Description
G	The CQA heterogeneous content network.
V, E, C	The node set, edge set, and contents of nodes.
e_{ij}	The relationship between two nodes (v_i, v_j) linked with each other.
$c_i \in C$	The text document associated with each node v_i.
d	The dimension of node embeddings.
λ	A meta path.
I_λ	A meta path instance is generated according to λ.
$\mathbb{C}_i^\lambda$	The meta path λ based context set of the target node v_i.
POS	The set of all positive node pairs generated through meta path based random walks.
NEG	The set of all negative sampling node pairs.

4 Proposed Model

In this section, we first demonstrate the overall framework of our proposed model. Then the details of our method will be described in Sections 4.2–4.4.

4.1 Overall Framework

We formulate the expert recommendation as a ranking task based on integrated node embedding by taking into account the partial ordering constraints contained in the CQA network. First, based on the released archive of the CQA website, we construct a heterogeneous content network that contains the textual content information of different nodes, unlike previous HINs that only contain nodes and edges. As shown in Fig. 1, the constructed heterogeneous CQA content network contains the user's past answering interactions (structure information) and the textual content information of each node. We can view the whole big Q&A data as two different networks in terms of structural proximity links and semantic content similarity links, as shown in Figs. 1b and 1c. Then, we design our learning model by using a HIN network embedding method equipped with hierarchical attention [42] to learn the topology-based embedding e^t, aiming to explore the proximity relationship between nodes and obtain network embeddings with more personalized preferences. Meanwhile, a meta-context aware skip-gram model is deployed to learn content-based embeddings e^c by taking advantage of the large amount of textual information contained in each node. Both learned embeddings aggregate structural or semantic information upon the sequences of context nodes generated by meta-path-guided random walks. Figs. 1e and 1f illustrate the two types of embeddings. The overall node embedding $e \in \mathcal{R}^d$ for each node is constructed by concatenating these two types of embeddings as shown below:

$$e = Concat(e^t, e^c) \tag{1}$$

where *Concat* () denotes the concatenation operation. The details of learning the topology-based network embedding and the content-based node embedding will be discussed in Sections 4.2 and 4.3.

After that, we use the ranking metric function σ to computer the probabilities of candidate answerers contributing the "best answers", which is defined as follows:

$$\sigma\left(e_q, e_a\right) = e_a^T e_q \tag{2}$$

Based on users' relative quality facts and the goal to learn the relative ranking of users to answer a specific question, we use the partial ordering constraints contained in the network to address the ranking and recommendation, the details will be discussed in Section 4.4. We train a model using positive and negative samples drawn from the network, and then implement recommendations based on our trained model when given a newly posted question with tags, title, and description.

4.2 Topology-Based Embedding via Hierarchical Attentive Meta Path

In this section, to explore the structural relationship and proximity properties in the heterogeneous CQA network, we deploy a hierarchical attention HIN embedding method [42] to learn the embedding of two target type nodes whose node type is *question* and *answerer*. Based on the previous definition and introduction of meta-path and path instance in Section 3, we will not repeat these two concepts but focus on explaining the hierarchical attention mechanism of the path instance attention layer and meta-path attention layer.

Given a CQA HIN $G = (V, E, C)$ and a meta path λ, nodes are connected by meta path instance I_λ. Many existing HIN embedding-based methods [43,44] have used the meta-path-based random walk to learn the overall proximity between nodes by maximizing the probability of predicting the node given its contextual nodes, but they do not distinguish meta-paths and the difference between path instances. As a result, the personalized preferences of nodes on meta paths and in path instances cannot be captured, and the proximity of nodes to be preserved is incomplete [42]. Therefore, to address the problem of indiscriminate learning, we deploy a hierarchical attention mechanism on the meta-path attention layer and the path-instance attention layer respectively to learn each node's personalized preference.

Combining MPBC embeddings by path instance attention layer. The path instance attention layer aims to learn the embedding E^λ according to the meta path λ by distinguishing the path instances. To learn this embedding, we not only rely on the information of the node itself but also collect the information of the context nodes connected by path instances. Meanwhile, we distinguish path instances based on the similarity of structural features. Specifically, the normalized meta path based on the adjacent vector A_i^λ is used as the structural feature representation of node v_i, and the nodes that connect to node v_i by path instances will be assigned an attention coefficient a^λ. Intuitively, those nodes that share similar structural features with node v_i will be highlighted by assigning a larger attention coefficient. Considering that the meta-path-based adjacent vector is usually high-dimensional and sparse, we first project the structure feature A_i^λ into d dimensional space:

$$A_i^\lambda = W^s A_i^\lambda A_j^\lambda = W^s A_j^\lambda \tag{3}$$

where $W^s \in \mathbb{R}^{N\times d}$ denotes the structural feature transformation matrix. Then the similarity calculation is performed to get the transformed feature similarity s_{ij}^λ between two nodes v_i and v_j based on meta path λ:

$$s_{ij}^\lambda = \frac{\left(A_i^\lambda\right)^T \cdot A_j^\lambda}{\left\|A_i^\lambda\right\| \cdot \left\|A_j^\lambda\right\|} \tag{4}$$

we calculate the path instance attention coefficients a^λ for each node and gathered the information from the meta-path-based context $\mathbb{C}$ to know the aggregated neighborhood embedding $\varkappa^\lambda$:

$$a_{ij}^\lambda = \frac{\exp(s_{ij}^\lambda)}{\sum_{k\epsilon\mathbb{C}_i^\lambda}(s_{ik}^\lambda)} \varkappa_{\mathbb{C}_i}^\lambda = \sigma\left(\sum_{j\epsilon\mathbb{C}_i^\lambda} a_{ij}^\lambda A_j^\lambda\right) \tag{5}$$

where a_{ij}^λ is the path instance attention coefficient of node v_j, $\mathbb{C}_i^\lambda$ is the context node set that connect to node v_i within path instances based on λ, σ represents the *tanh* activation function. Finally, the aggregated neighborhood embedding $\varkappa_{\mathbb{C}_i}^\lambda$ and the node's feature embedding A_i^λ are concatenated together to generate the final meta path based embedding e_i^λ:

$$e_i^\lambda = \text{Concat}\left(\varkappa_{\mathbb{C}_i}^\lambda, A_i^\lambda\right) W^\lambda \tag{6}$$

where $W^\lambda \in \mathbb{R}^{2d\times d}$ denotes a learnable linear transformation matrix. In summary, the vector e_i^λ learned from the path instance attention layer is a composite representation which contains its own node structural information and the meta path based global structural information.

Topology-based comprehensive embedding by meta path attention layer. The meta-path attention layer aims to learn the comprehensive embedding E^t by weighted combining the meta-path-based embeddings learned from the path instance attention layer to capture the personalized preference on meta paths of each node. A randomly initialized meta-path preference vector $r_i \in \mathbb{R}^{1\times k}$ for each node v_i is introduced to perform the meta-path attention mechanism, which will be updated during the training process. To get the personalized attention coefficients on meta path λ for node v_i, we first measure the similarity between r_i and transformed meta path based embedding e_i^λ learned from the previous path instance attention layer, and then we calculate the meta path attention coefficient η_i^λ:

$$\chi_i^\lambda = \frac{{r_i}^T \cdot \sigma\left(W^r e_i^\lambda + b_r\right)}{\|r_i\| \cdot \left\|\sigma\left(W^r e_i^\lambda + b_r\right)\right\|} \eta_i^\lambda = \frac{exp\left(\chi_i^\lambda\right)}{\sum_{k=1}^{\mu} exp\left(\chi_i^k\right)} \tag{7}$$

where $W^r \in \mathbb{R}^{d\times k}$ is the transformation matrix, b_r denotes the bias parameter, σ denotes the *tanh* activation function, and μ denotes the number of meta paths. We get the final topology-based comprehensive embedding e^t by weighted merging operation:

$$e_i^t = \sum_{\lambda=1}^{\mu} \eta_i^\lambda e_i^\lambda \tag{8}$$

In summary, the hierarchical attention mechanism deployed from the above two different perspectives enables our model to learn the discriminate scores of the structural proximity between each node and its neighbor nodes connected by path instance and to highlight the node with the most topology-related information on the meta-path.

We learn the parameters in a task-specific semi-supervised environment by exploiting more information contained in the CQA network. Specifically, we integrate the representations of nodes into a node classifier (implemented with a full connection layer with *softmax* function) to infer the probability of node v_i on label l. The loss function aims to minimize the Cross-Entropy loss between the ground truth and the predictions:

$$\boldsymbol{L_t} = -\sum\nolimits_{i}^{|V|} \sum\nolimits_{l=1}^{L} \ell_{il} \ln \left(\boldsymbol{P} \left(e_i^t \right)_l \right) \tag{9}$$

where $\boldsymbol{P}\left(e_i^t\right)_l \in \{\mathbf{0}, \mathbf{1}\}$ is the predicted result of node v_i on label l, and $\ell_{il} \in \{\mathbf{0}, \mathbf{1}\}$ is the ground truth of node v_i on label l. We can further employ several node classifiers for different types of nodes. The parameters of hierarchical attention layers are shared and trained by multiple classifiers.

4.3 Content-Based Embedding via Meta-Context Aware Skip-Gram Model

Our proposed model ER-HCNE is anticipated to integrate typical heterogeneous information, such as the type information of different nodes, node textual content, etc. In Section 4.2, we use a meta-path based random walk method to obtain topology-based embeddings, which exploits node type information by restricting transitions between nodes by meta-paths. However, the large amount of textual content information contained in nodes and textual relevance between different nodes have not been fully explored. Therefore, in this section, we focus on how to utilize node textual content information to obtain the content-based embedding e^c, aiming to better understand node text semantics and text correlation between nodes, thereby enhancing node representation. Apart from previous studies which obtain node's content embedding by directly using node own textual content through a feature extractor (i.e., an LSTM-based model or a CNN-based model), we designed a new method to obtain contextual embedding by a meta-context aware skip-gram model with negative sampling. More specifically, we first encode each node textual content c_i into a vector as the textual feature representation of node v_i and then capture the semantic relevance by maximizing the probability of predicting the target node given its related nodes.

Encode node textual content. To encode each node textual content into fixed length feature embedding, we introduce the pre-trained deep bidirectional Transformers based model BERT [30]. The final output of $BERT_{concat}$ encoder is denoted by $e^{own} \in \mathbb{R}^{l \times d}$, where l is the number of terms of a node textual content. According to *Definition 1*, we take different components to construct the textual content of different types of node c_i. Specifically, when the node type is the question, answerer, and tag, the node content is the combination of the question's title and body, the title combination of questions for which the user has provided the best answer, and the term of the tag, respectively. Given the node content $c_i = \{c_i^1, c_i^2, \ldots, c_i^l\}$, where c_i^j is the j-th term in c_i and l is the number of terms in c_i. We take each node content c_i as input for $BERT_{concat}$:

$$e^{own} = BERT_{concat}([CLS], c_i^1, c_i^2, \ldots, c_i^l, [SEP]) \tag{10}$$

where $[CLS]$ and $[SEP]$ are two special tokens that mark the beginning and the end of a sentence.

The encoder used in BERT was introduced from Transformer [29] which is a multi-head self-attention deep learning architecture. Compare with CNNs or LSTMs adopted in previous works [8,18], BERT generates dynamic contextual word embedding, which is a function of the entire input text by taking into account the word dependencies and sentence structures. Given an input node text sequence of length l, it will go through several layers of encoders to generate l vector representations of length 768 at each layer according to each token. In particular, we would like to get a single vector representation of the whole textual sequence. To achieve this, we average the last four hidden layers of each token to produce a single 768-length vector and project it to a 128-dimensional vector space as the final node content feature representation e^{own}.

Due to the space limitations, we will no longer introduce the Transformer encoder blocks in detail since many previous works have been discussed and the mathematical details can be found in [29,30].

Combing contextual node information on MPBC. So far, the individual textual information-based embedding e^{own} for each node is generated. However, each node is not isolated in the network, and it is proved to be very effective and necessary to combine the content of contextual nodes to understand the semantics of the current node [36,45]. To integrate context node information and explore the degree of influence by different context nodes, we first calculated the attention coefficient for each pair based on the text similarity evaluation:

$$similarity(v_i, v_j) = \frac{\left(e_i^{own}\right)^T \cdot e_j^{own}}{\left\|e_i^{own}\right\| \cdot \left\|e_j^{own}\right\|} \tag{11}$$

and then a weighted sum representation is constructed to strengthen the node content embedding:

$$e_i^c = \tanh\left(e_i^{own} + \sum\nolimits_{j\epsilon\mathbb{C}_i} \frac{\exp(similarity(v_i, v_j))}{\sum_{k\epsilon\mathbb{C}_i}(similarity(v_i, v_k))} e_j^{own}\right) \tag{12}$$

Skip-gram with negative sampling for Content-based Embedding learning. We treat the node sequence sampling from meta-path-guided random walks as a sentence and each node as a word in the sentence. Our goal is to maximize the following probability:

$$g\left(v_i\right) = \prod\nolimits_{i=1}^{|V|} p\left(v_i|context(v_i)\right) = \prod\nolimits_{i=1}^{|V|} \prod\nolimits_{v_j\in\{\mathbb{C}_i\}} \sigma\left(e_i^c \cdot e_j^c\right) \tag{13}$$

where $p\left(v_i|context(v_i)\right)$ defines the probability of predicting the node v_i given the contextual nodes $context(v_i)$. e_i^c and e_j^c is the node content embedding of node v_i and v_j, respectively. $\mathbb{C}_i$ is the context set of the target node v_i, and $\sigma(x) = 1/(1 + \exp(-x))$ represents the logistic function.

In order to maximize the conditional probability between e_i and e_j and relieve the expensive computational cost of *softmax* function for all nodes, the negative sampling [46] strategy is adopted to approximate the objective function as follows:

$$\boldsymbol{L_c} = \sum\nolimits_{(v_i,v_j)\in POS} \log\left(\sigma\left(e_i^c \cdot e_j^c\right)\right) + \sum\nolimits_{(v_i,v_j')\in NEG} \log\left(-\sigma\left(e_i^c \cdot e_{j'}^c\right)\right) \tag{14}$$

where POS is the set of positive node pair samples, and NEG is the set of negative node pair samples. During the training process, the positive samples are generated by retrieving the neighbors in the meta path guided random walks with a length-w sliding window, while a negative sample v_j' can be randomly drawn from the distribution $p(v_j'|v_i)$ for each neighbor.

4.4 Joint Learning and Ranking Loss for Recommendation

In Sections 4.2 and 4.3, we describe how to characterize the nodes in CQA heterogeneous networks more accurately from two perspectives by exploring the proximity relationship and semantic relevance. With the above formulations, we optimize the following joint objective function, which is a weighted combination of the topology-based embedding loss and the content-based embedding loss:

$$\boldsymbol{L_{emb}} = \gamma * \boldsymbol{L_t} + (1-\gamma) * \boldsymbol{L_c} \tag{15}$$

where $\gamma \in [0, 1]$ is a parameter to balance the importance of the two loss functions. When γ increases, more structural information will be considered to characterize the nodes in the network, and conversely, textual content information will account for a larger share. According to Eq. 1, the final learned representation e of each node can retain the network structural proximity and the textual content relevance of nodes and will be used as the features of questions and users for the following ranking and recommendation modules. According to our observations on the form of the question thread in the Stack Overflow Q&A community, we formulate expert recommendations as a ranking task after learning the compound node embedding e. Specifically, two partial ordering constraints contained in question sessions can be observed: (1) Users who provide answers to the question q_i should have a higher score than those who did not. (2) The answerer who provided the "best answer" to the question q_i should have a higher score than other answerers in the same question thread. Therefore, instead of expecting a specific probability, we apply a raking triple loss function to learn the relative ranking of users to answer a certain question $q_i \in Q$ in a question session. The ranking loss is shown as follows by translating the above two constraints:

$$\mathcal{L}_{rank}(\Theta) = \sum_{q_i \in Q} \left\{ \sum_{\substack{a^+ \in S_i \\ a^- \notin S_i}} \left(\sigma\left(e_{q_i}, e_{a^+}\right) - \sigma\left(e_{q_i}, e_{a^-}\right)\right) + \sum_{\substack{a^+ \in S_i \\ a^{best} \in S_i}} \left(\sigma\left(e_{q_i}, e_{a^{best}}\right) - \sigma\left(e_{q_i}, e_{a^+}\right)\right) \right\} \tag{16}$$

where s_i denotes the list of answerers who provide answers to the question q_i, and e_{q_i} denotes the compound embedding of q_i. The variables e_{a^+}, e_{a^-}, and $e_{a^{best}}$ represent the embeddings of the positive answerers, negative answerers, and the best answerers, respectively. Maximizing $\mathcal{L}_{rank}(\Theta)$ is to encourage reducing the distance between the question and the actual answerer while capturing the relative quality differences between different answerers.

5 Experimental Evaluation

In this section, we present experiments to evaluate the performance of our proposed method for the expert recommendation task.

5.1 Datasets

Stack Overflow is one of the most popular programming Q&A communities used by millions of programmers. Every achieved question includes the title, body, and several tags that identify the area of knowledge involved and has only one best answer which usually with the highest number of votes. The user who provides the best answer is selected as the best answerer. The profile of each user can be found in detail on the page too, including the list of questions for which they have provided the best answer and tags representing the user's expertise and research field. We conduct our experiments on a real-world CQA data dump collected from the Stack Overflow websites which was published on March 14, 2017, covering from July 2008 to March 2017, and available online[1]. Specifically, to better evaluate the performance, we constructed three datasets as described in [47] from three different fields: *Java*, *Python*, and *C#*. Each selected question for evaluation is an archived question with an acceptable answer (oot note text for 1i.e., the best answer) and at least 3 answers. Each question can be labeled according to its relevant tag, and each author can be labeled according to his/her field of research, which can be found in his/her profile. In total, we collected 29,511 questions, each of which was an archived question with one best answer. However, the number of valid questions for evaluation in the

[1] https://archive.org/details/stackexchange

three datasets is only 4661, 2609, and 6492, respectively. For each test question q, we create a candidate answerer set consisting of 3 negative answerers randomly selected from the top 10% of the most active users (excluding the answerers to the current question), as well as all positive answerers who are in the answerer list of q. The best answer provider is considered to have more expertise than the other answerers in a question thread. Our goal is to find users' relative quality rankings rather than scoring users' knowledge and expertise of a certain question. The statistical details of datasets are shown in Table 2.

Table 2: Statistics of datasets

Tag	# of Questions (with best answerer)	# of Valid Questions	# of active Users	# of edges	Node type T_V	Meta-path
Java	11,940	4,661	2,977	20,749	Question(Q)	AQA
Python	5,977	2,609	1,983	11,844	Answerer(A)	AQTQA
C#	11,594	6,492	4,903	32,615	tag(T)	

5.2 Evaluation Metrics

We use three widely used evaluation metrics to evaluate the performance of our proposed model: Mean Average Precision (MAP), Precision at K ($p@K$) and Mean Reciprocal Rank (MRR).

(1) MRR. The MRR denotes the average inverse of the rank of the correct answerer:

$$MRR = \frac{1}{|N|} \sum\nolimits_{q \in Q} \frac{1}{rank_j} \tag{17}$$

where N denotes the total number of questions and $rank_j$ is the sequence location of the ground truth answerer who provided the accepted answer.

(2) $p@K$. The $p@K$ reports the proportion of predicted samples where the ground truth answerers appear in the ranked top-K result. For example, $p@5$ reports the percentage of ground truth answerers appearing in the top 5 search results. A special example is $P@1$, which aims to calculate the percentage of times the system ranks the correct answerer at the top.

(3) *Mean Average Precision* (MAP): The MAP reports the overall retrieval quality score, which is the mean of the average precision scores for each question.

5.3 Baselines and Experimental Settings

Baselines We use the following methods for experiment comparison:

Content-Based Method

(1) ***LDA*** [36] is a three-level hierarchical Bayesian model, which relies on word co-occurrence as well as having the ability of semantic understanding. In our experiments, we construct a user's profile by connecting all the questions answered by the user. For training, the Gibbs-LDA++ [37] with topic size K=100 is applied, and we set the LDA hyper-parameters to $\alpha = 0.5$ and $\beta = 0.1$.

(2) ***QR-DSSM*** [48] was proposed by Azzam et al. by directly applying the Deep Semantic Similarity Model [26]. In our experiment, two fully connected DNNs which contain two hidden

layers with 300 nodes in each are applied to learn the feature vectors, and cosine similarity calculations are conducted after the output layer vectors. The iterations number is 100 and the learning rate is 0.02.

Topology-Based Method

(3) ***Node2vec*** [32] is a homogeneous network embedding method that extends *DeepWalk* [31] by broadening the definition of network neighborhood and designing a biased random walk strategy to explore more diverse node representations. In our experiment, the length of sampled paths is 5, the size of embedding is 128, and the number of walkers is 10.
(4) ***Metapath2vec*** [33] is one of the state-of-the-art network embedding methods for HIN, which uses meta-path-guided random walks to build heterogeneous neighborhoods for each node, and then uses the Skip-Gram model to learn the node embeddings. We leverage the meta paths AQTQA and AQA in this method and the dimension is set to 128.

Combined Method

(5) ***NeRank (Metapath2vec+LSTM)*** [21] is designed to jointly learn the representation of different entities via the meta path-based heterogeneous network embedding strategy and a long short-term memory(LSTM) model. A CNN-based scoring function is used to calculate the ranking score. However, it does not consider the weight of different meta paths and only utilizes the question content. We adopted the default experimental setting mentioned in [21] with a meta-path length of 13, node coverage of 20, the window size of the Skip-gram model of 4, and the dimension of learning embeddings set to 128 for fair competition.
(6) ***NW (Node2vec+Word2vec)*** [20] jointly considers the graph-based similarity and text-based similarity to address the expert finding problem. According to the experiment result in [20],we employ Node2Vecto map users to a vector space and compute the similarity between the question's asker and candidate experts to obtain the graph-based similarity score. The text-based similarity was obtained by computing the similarity between the question content and the candidate expert's profile.

Among the above six baselines, *LDA* is the traditional algorithm that learns the question and user's representation based on bag-of-words contents and relies on manual feature engineering. In contrast, *QR-DSSM* is based on deep learning to automatically learn the feature vector for text without human intervention or assistance. These two content-based methods focus on the representation learning of question contents and user profile to construct a semantic feature space. While topology-based methods focus on learning node embeddings by taking advantage of the network structural properties rather than text semantic features. Specifically, *Node2vec* learns representations of questions and users through a random walk strategy, while *Metapath2vec* generates walk sequences guided by predefined meta-paths. In addition, we use two advanced combined methods to verify the effectiveness of our model. *NeRank* is one of the competitive methods by integrating a meta-path-based skip-gram model and an LSTM-based question content encoder. *NW* is proposed in 2021 by employing Node2Vec and Word2vec to learn graph-based similarity and text-based similarity. Unlike the previous studies, our proposed model ER-HCNE exploits the textual content of various types of nodes instead of only using the content of the question node. Moreover, we use the Transformer based representation encoder and attention mechanism to obtain contextual content-based embedding. In addition, we optimize meta-path-based random walks through a two-layer attention mechanism-based model.

Experimental Settings Our proposed model ER-HCNE is implemented with PyTorch. The embedding dimension $d = 128$. We set the learning rate to 0.0005 and the batch size to 512. The window size of the Skip-gram model is set to 5 and the size of the negative samples is set to 3. The walks are generated from meta paths "AQA" and "AQTQA" with a default length of 12 and the number of walks per node is set to 10. In the content-based embedding learning module, we use the English uncased BERT-Base model released by Google [30], which has 12 layers and 768 hidden states. The maximum length of question content and answerer content is set to 300 tokens. We labeled about one-third of the nodes in each of the two different node types for training. The model parameters are randomly initialized, and an Adam optimizer is employed for optimization. For the baseline methods, we use the code provided by the authors.

5.4 Experimental Results and Analysis

In this section, we report the experimental results and analyses of the effectiveness and efficiency of our proposed model. The three metrics mentioned before (MRR, P@K, and MAP) are applied to evaluate the ranking performance.

A. Performance Comparison with Different Baseline Methods

Although the scales of the three datasets are different, the performance metrics show similar trends. Therefore, we first take the average performance of the three datasets for overall analysis and comparison in Table 3. From the results, we can make the following observations:

- Combined methods exhibit much better performance than topology-based methods or content-based methods. This suggests that it is very effective to combine content analysis and structure exploration for representation learning in CQA networks, which can greatly alleviate the sparsity of structural or textual data by learning from different perspectives.
- Our proposed model ER-HCNE significantly and consistently outperforms all baselines in our datasets on all metrics. For example, ER-HCNE outperforms the competitive methods NeRank and NW by 12.8% and 13.7% on MAP, respectively. The main reason is that ER-HCNE considers the personalized preferences of nodes on meta-paths and exploits the textual content of both question and answerer nodes.
- Moreover, since there is only one best answerer per question, it is very challenging to rank among thousands of candidates for our task. However, ER-HCNE can still achieve 34.3% precision of the P@1 metric, which indicates that each test question will be answered if we route it to the top 3 users on average, while NeRank and NW required at least 4 users.
- Although both are content-based methods, the performance of deep learning-based QR-DSSM is far superior to traditional LDA methods, which shows that mining deep text semantics is very important in content-based methods.
- Between the two combined methods, NeRank outperforms NW in most cases, but it is worth noting that NW slightly outperforms NeRank on all metrics when the training data is small. The main reason may be that walks in Node2vec are more flexible than Metapath2vec, and NW combines structural and textual features to represent each node, while NeRank only encodes the textual content of the question nodes. Therefore, NW can capture more comprehensive information than NeRank when the amount of training data is small. However, as the number of data increases, the advantages of meta-path-based HIN embedding methods prevail.
- The performances of Node2vec and QR-DSSM are both acceptable and very close. The best values of the MAP metrics can reach 51.1% and 50.1%, respectively. This shows that the modeling effect of using network topology information or textual content information is effective.

Table 3: Comparison of different methods for expert recommendation

Model type	Model name	# Training: 50%				# Training: 70%				# Training: 90%			
		P@1	P@5	MRR	MAP	P@1	P@5	MRR	MAP	P@1	P@5	MRR	MAP
Content-based	LDA	0.068	0.189	0.106	0.289	0.076	0.263	0.221	0.311	0.105	0.289	0.251	0.345
	QR-DSSM	**0.102**	0.301	0.278	**0.392**	0.134	0.433	0.331	0.419	**0.182**	0.485	0.462	0.501
Topology-based	Node2vec	0.098	0.304	0.299	0.389	0.138	0.479	0.429	0.425	0.179	0.501	0.489	0.511
	Metapath2vec	0.115	0.415	0.422	0.421	0.154	0.501	0.438	0.488	0.199	0.515	0.525	0.526
Combined method	NeRank	0.177	0.468	0.479	0.562	**0.263**	**0.566**	**0.521**	**0.621**	**0.321**	**0.626**	**0.566**	**0.631**
	NW	**0.181**	**0.472**	**0.502**	**0.577**	0.198	0.522	0.511	0.617	0.254	0.568	0.534	0.626
ER-HCNE	w/o structure	0.121	0.342	0.349	0.432	0.145	0.441	0.437	0.492	0.195	0.494	0.481	0.524
	w/o content	0.135	0.335	0.352	0.428	0.163	0.471	0.492	0.556	0.224	0.511	0.529	0.559
	ER-HCNE$_{LSTM}$	0.194	0.486	0.496	0.579	0.286	0.577	0.538	0.634	0.331	0.664	0.576	0.689
	ER-HCNE-MP	0.191	0.498	0.483	0.587	0.267	0.568	0.527	0.629	0.316	0.651	0.569	0.677
	ER-HCNE	**0.202**	**0.522**	**0.511**	**0.598**	**0.311**	**0.671**	**0.573**	**0.647**	**0.343**	**0.685**	**0.589**	**0.712**

- As the training data increases, the performance of all methods improves. It shows that the more nodes, edges, and content involved, the more comprehensive and specific the learning of semantic correlations and structural relationships will be.

B. Analysis of Different Components in ER-HCNE

Fig. 3 illustrates the performance comparison between the variants of ER-HCNE and the start-of-the-art baseline method, we can observe that:

- Our model still performs better than the baseline model, NeRank, without using the BERT sequence encoder (ER-HCNE$_{\mathrm{LSTM}}$) or attention mechanism (ER-HCNE-MP). This indicates that incorporating the answerer's content could enhance the representation and improve the ranking performance.
- Compared with ER-HCNE, the performance of methods without topology-based embedding (w/o structure) or content-based embedding (w/o content) drops sharply by 27.88% and 25.4% on the P@5 metric, respectively. This indicates that our combined model is efficient and can greatly improve the matching and ranking performance.

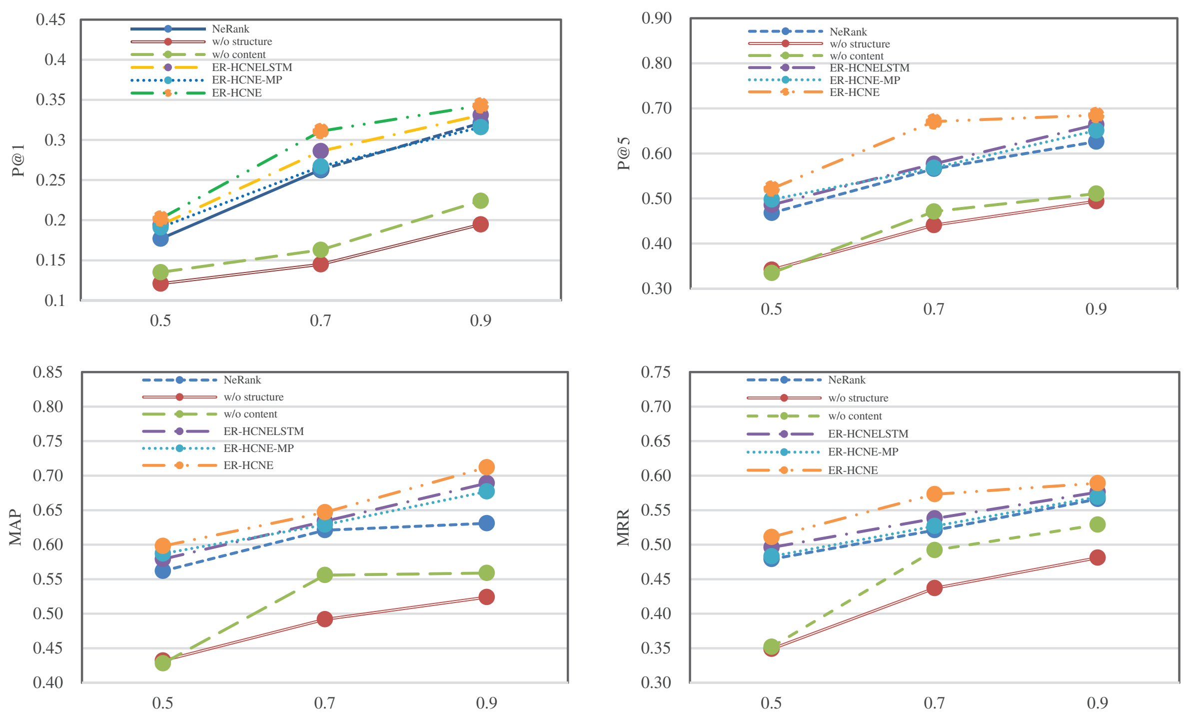


Figure 3: Performance comparison between the variants of ER-HCNE and the best baseline method

C. Effectiveness of Topology-Based Embedding via Hierarchical Attentive Meta Path

In order to verify the effectiveness of the hierarchical attentive meta path mechanism, we compare ER-HCNE with three of its variants: (i) **ER-HCNE-MP**: We learn the topology-based embeddings via metapath2vec [39] without attention mechanism. (ii) **ER-HCNE-DW**: Instead of the meta-path-based HIN embedding method, we adopt DeepWalk [31] model to learn the network embeddings. (iii) **ER-HCNE-Node2vec**: Another node embedding model **Node2vec** [32] is employed to learn topology-based embeddings.

Fig. 4 shows the experimental results. We observe that ER-HCNE outperforms its three variants on all metrics over the three datasets. In addition, ER-HCNE-MP performs slightly better than the other two methods. Specifically, ER-HCNE-MP, ER-HCNE-Node2vec, and ER-HCNE-DW decreased by 4.86%, 9.73%, and 10.01% in MAP metrics compared with ER-HCNE, respectively. This is mainly due to the following reasons: First, CQA networks are heterogeneous, while DeepWalk and Node2vec are designed for homogeneous networks with limited ability to explore complex heterogeneous information and relationships contained in different nodes. Second, even without the hierarchical attention mechanism, the meta path-based method ER-HCNE-MP can also leverage the node type information and semantic information of different entities, thereby still performing better than ER-HCNE-DW and ER-HCNE-Node2vec. Third, modeling the personalized preference on both meta path and path instances by the hierarchical mechanism can help learn better network embeddings of different entities in CQA and boost the ranking performance.

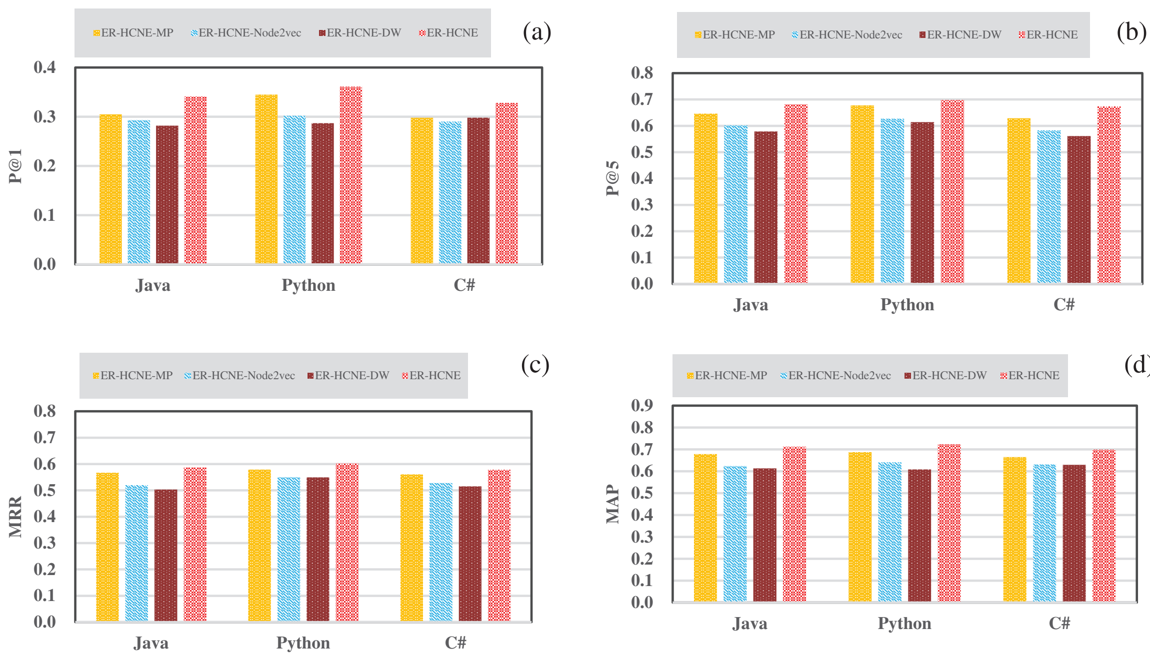


Figure 4: Performance comparison of methods employing different topological embedding strategies

D. Effectiveness of Combining Node Content Embedding and Meta-aware Contextual Embedding

To verify the effectiveness of our choices of adopting BERT to encode each node's textual content and combing contextual node information on MPBC to obtain more meaningful node embeddings, we design three variant models based on ER-HCNE as follows:

1. ER-HCNE$_{LSTM}$: We replaced the BERT encoder with an LSTM-based encoder, taking 300-dimensional word embeddings by Glove as the input.
2. ER-HCNE$_{CNN}$: We replaced BERT with another commonly used encoder, a CNN-based encoder. We still take 300-dimensional word embeddings by Glove as the input.
3. ER-HCNE$_{own}$: To demonstrate the effectiveness of combining meta context-aware embedding into the node own content embedding, we only keep the individual textual information-based embedding e^{own} as the input content-based embedding for each node.

The performance comparison between these three variants and ER-HCNE is shown in Table 4. We can conclude that:

- ER-HCNE$_{LSTM}$ still performs better than NW and NeRank even without using BERT encoder by 10.1% and 9.24% on MAP evaluation metrics. This shows that the performance improvement of our proposed model is not only due to the choice of the pre-trained sequence encoder BERT. However, the use of BERT does enhance the representation of textual features.
- ER-HCNE significantly outperforms ER-HCNE$_{own}$ on all metrics in three different datasets. Without combining the contextual node information learned on MPBC, ER-HCNE$_{own}$ drops 10.07%, 7.97%, and 12.78% on p@5, MRR, and MAP metrics on average compared with ER-HCNE, respectively. This indicates that in complex CQA networks, it is very necessary to collect semantic information about adjacent nodes and learn textual relevance between nodes.
- We can also observe that ER-HCNE$_{LSTM}$ and ER-HCNE$_{CNN}$ exceed ER-HCNE$_{own}$ in all metrics but are slightly lower than ER-HCNE. This shows that different encoders do have an impact on performance, but it's not the most critical factor. Combining other node information learned from meta-paths with the textual information of the nodes themselves is a key factor in determining the representation performance.

Table 4: Performance comparison of different content encoding methods

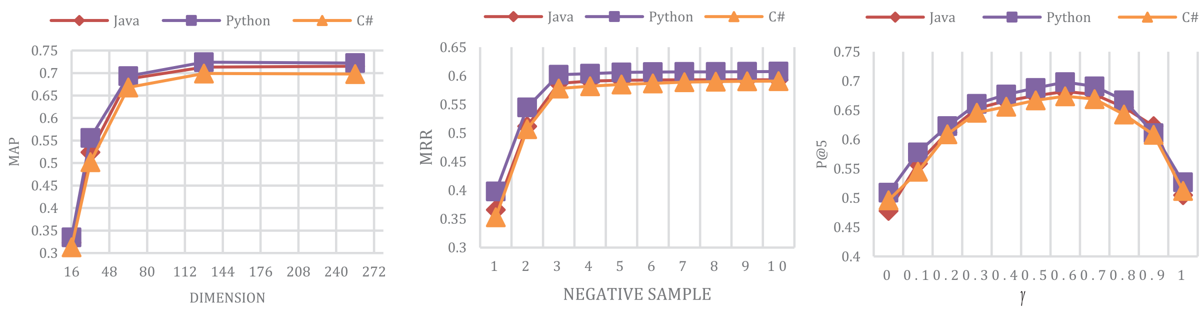
Datasets	Java				Python				C#			
	P@1	P@5	MRR	MAP	P@1	P@5	MRR	MAP	P@1	P@5	MRR	MAP
ER-HCNE$_{LSTM}$	0.326	0.635	0.570	0.674	0.341	0.692	0.585	0.699	0.327	0.666	0.573	0.695
ER-HCNE$_{CNN}$	0.317	0.608	0.547	0.644	0.326	0.656	0.567	0.648	0.325	0.643	0.556	0.649
ER-HCNE$_{own}$	0.312	0.595	0.539	0.619	0.318	0.632	0.537	0.618	0.316	0.621	0.549	0.626
ER-HCNE	**0.341**	**0.682**	**0.587**	**0.713**	**0.361**	**0.698**	**0.602**	**0.724**	**0.328**	**0.674**	**0.578**	**0.699**

E. Parameter Study

In this section, we discuss three essential parameters in ER-HCNE, which are the size of embedding dimensions, the size of the negative samples, and the balance weight γ (in Section 4.4 Joint learning and ranking loss for the recommendation). We vary the size of embedding dimensions from 16 to 256. Fig. 5a shows the MAP scores of ER-HCNE over different datasets with different embedding sizes. When the size of the dimension increases, the performance improves and peaks at 128. After that, even though the embedding dimension continued to increase, the performance trend stabilized without much improvement. Therefore, we choose the embedding dimension $d = 128$ in our experiments.

Fig. 5b shows the trends of the metric MRR with the negative sample size varying from 1 to 10 on the three datasets. We can see that all three metrics increase rapidly when the negative sample size goes from 1 to 3, the reason should be that our model needs enough negative samples to identify correct answerers. When the number of negative samples increased to more than 3, the growth of all metrics became very slow. Therefore, to reduce the cost of training and maintain high performance, we take the value of the negative sample as 3.

The effect of the balance weight γ is shown in Fig. 5c. The performance of ER-HCNE initially increases gradually with increasing γ but starts to decrease when γ exceeds 0.6. Note that if $\gamma = 0$, only textual content information is considered, and when γ grows to 1, our model degenerates to a topology-based method. Furthermore, we can observe that the performance drops sharply when γ increases from 0.9 to 1.0, which indicates the importance of textual content information.



(a) (b) (c)

Figure 5: Parameter study

6 Conclusion

CQA websites have got rapid development and the expert recommendation task has attracted considerable attention in recent years. In this paper, we approach the expert recommendation problem from the perspective of learning ranking metric embeddings by exploring various heterogeneous information and exploiting the relative quality ranking of users in question sessions. The key to our proposed model ER-HCNE is that we jointly consider the node proximity relations as well as the textual content correlation between meta-path-based adjacent nodes to learn more comprehensive representation in heterogeneous social networks. We conduct extensive experiments on Stack Overflow datasets and compare the results with several state-of-the-art models. The evaluation results of different metrics show that our proposed framework may help improve the efficiency of question-solving and bring higher satisfaction and experience to users in CQA. The improvement in the efficiency of the proposed work is due to the feature extraction from the text based on the text vector. In future work, we hope to improve the quality of node embeddings by introducing more QA features or non-QA features, and we would like to explore more efficient ways to mine connections and relationships between various entities in CQA heterogeneous information networks.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. Moutidis and H. Williams, "Community evolution on stack overflow," *PloS One*, vol. 16, no. 6, pp. 1–14, 2021.
[2] K. Balog, L. Azzopardi and M. De Rijke, "Formal models for expert finding in enterprise corpora," in *Proc. of the 29th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Seattle, Western Australia, pp. 43–50, 2006.
[3] B. Li, I. King and M. R. Lyu, "Question routing in community question answering: Putting category in its place," in *Proc. of the 20th ACM Int. Conf. on Information and Knowledge Management*, Scotland, UK, pp. 2041–2044, 2011.

[4] Z.-M. Zhou, M. Lan, Z.-Y. Niu and Y. Lu, “Exploiting user profile information for answer ranking in cqa,” in *Proc. of the 21st Int. Conf. on World Wide Web*, Lyon, France, pp. 767–774, 2012.
[5] J. Liu, Y.-I. Song and C.-Y. Lin, “Competition-based user expertise score estimation,” in *Proc. of the 34th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Beijing, China, pp. 425–434, 2011.
[6] F. Riahi, Z. Zolaktaf, M. Shafiei and E. Milios, “Finding expert users in community question answering,” in *Proc. of the 21st Int. Conf. on World Wide Web*, Lyon, France, pp. 791–798, 2012.
[7] Z. Zhao, Q. Yang, D. Cai, X. He and Y. Zhuang, “Expert finding for community-based question answering via ranking metric network learning,” in *Proc. of the Twenty-Fifth Int. Joint Conf. on Artificial Intelligence*, New York, USA, 16, pp. 3000–3006, 2016.
[8] J. Wang, J. Sun, H. Lin, H. Dong and S. Zhang, “Convolutional neural networks for expert recommendation in community question answering,” *Science China Information Sciences*, vol. 60, no. 11, pp. 1–9, 2017.
[9] X. Liu, W. B. Croft and M. Koll, “Finding experts in community-based question-answering services,” in *Proc. of the 14th ACM Int. conf. on Information and knowledge management*, Breman, Germany, pp. 315–316, 2005.
[10] Y. Tian, P. S. Kochhar, E. P. Lim, F. Zhu and D. Lo, “Predicting best answerers for new questions: An approach leveraging topic modeling and collaborative voting,” in *Int. Conf. on Social Informatics*, Kyoto, Japan, pp. 55–68, 2013.
[11] H. Li, S. Jin and L. Shudong, “A hybrid model for experts finding in community question answering,” in *2015 Int. Conf. on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Xi’an, China, pp. 176–185, 2015.
[12] A. M. Elkahky, Y. Song and X. He, “A multi-view deep learning approach for cross domain user modeling in recommendation systems,” in *Proc. of the 24th Int. Conf. on World Wide Web*, Florence, Italy, pp. 278–288, 2015.
[13] Z. Liu and B. J. Jansen, “Analysis of question and answering behavior in question routing services,” in *CYTED-RITOS International Workshop on Groupware*, Berlin, Germany, pp. 72–85, 2015.
[14] B. Mathew, R. Dutt, S. K. Maity, P. Goyal and A. Mukherjee, “Deep dive into anonymity: Large scale analysis of quora questions,” in *Int. Conf. on Social Informatics*, Doha, Qatar, pp. 35–49, 2019.
[15] G. Wang, K. Gill, M. Mohanlal, H. Zheng and B. Y. Zhao, “Wisdom in the social crowd: An analysis of quora,” in *Proc. of the 22nd Int. Conf. on World Wide Web*, Rio de Janeiro, Brazil, pp. 1341–1352, 2013.
[16] G. Zhou, K. Liu and J. Zhao, “Monolingual-based translation model for Question Routing,” in *Chinese Conf. on Pattern Recognition*, Beijing, China, pp. 630–637, 2012.
[17] L. Wang, B. Wu, J. Yang and S. Peng, “Personalized recommendation for new questions in community question answering,” in *Proc. of the 2016 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, Davis California, USA, pp. 901–908, 2016.
[18] H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang *et al.,* “Community-based question answering via heterogeneous social network learning,” in *Proc. of the AAAI Conf. on Artificial Intelligence*, Arizona, USA, vol. 30, pp. 1–14, 2016.
[19] J. Sun, J. Zhao, H. Sun and S. Parthasarathy, “EndCold: An End-to-End framework for cold question routing in community question answering services,” in *Proc. of the Twenty-Ninth Int. Joint Conf. on Artificial Intelligence*, Yokohama, Japan, pp. 3244–3250, 2020.
[20] N. Ghasemi, R. Fatourechi and S. Momtazi, “User embedding for expert finding in community question answering,” *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 4, pp. 1–16, 2021.
[21] Z. Li, J.-Y. Jiang, Y. Sun and W. Wang, “Personalized question routing via heterogeneous network embedding,” in *Proc. of the Thirty-Third AAAI Conf. on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conf. and Ninth AAAI Symp. on Educational Advances in Artificial Intelligence*, Honolulu, Hawaii USA, pp. 192–199, 2019.
[22] S. Momtazi and F. Naumann, “Topic modeling for expert finding using latent Dirichlet allocation,” *Wiley Interdisciplinary Reviews: Data Mining Knowledge Discovery*, vol. 3, no. 5, pp. 346–353, 2013.

[23] G. Zhou, J. Zhao, T. He and W. Wu, "An empirical study of topic-sensitive probabilistic model for expert finding in question answer communities," *Knowledge-Based Systems*, vol. 66, no. 5, pp. 136–145, 2014.

[24] L. Yang, M. Qiu, S. Gottipatti, F. Zhu, J. Jiang *et al.,* "Cqarank: Jointly model topics and expertise in community question answering," in *Proc. of the 22nd ACM Int. Conf. on Information & Knowledge Management*, California, USA, pp. 99–108, 2013.

[25] A. Azzam, N. Taziand and A. Hossny, "Text-based question routing for question answering communities via deep learning," in *Proceedings of the Symposium on Applied Computing*, Marrakech, Morocco, pp. 1674–1678, 2017.

[26] P. S. Huang, X. He, J. Gao, L. Deng, A. Acero *et al.,* "Learning deep structured semantic models for web search using clickthrough data," in *Proc. of the 22nd ACM Int. Conf. on Information & Knowledge Management*, California, USA, pp. 2333–2338, 2013.

[27] S. MacAvaney, A. Yates, A. Cohan and N. Goharian, "Cedr: Contextualized embeddings for document ranking," in *Proc. of the 42nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Paris, UK, pp. 1101–1104, 2019.

[28] Z. Dai and J. Callan, "Deeper text understanding for IR with contextual neural language modeling," in *Proc. of the 42nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Paris, UK, pp. 985–988, 2019.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[30] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ACL Anthology*, vol. 1, pp. 1–12, 2018.

[31] B. Perozzi, R. Al-Rfouand and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. of the 20th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Newyork, USA, pp. 701–710, 2014.

[32] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Newyork, USA, pp. 855–864, 2016.

[33] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan *et al.,* "Line: Large-scale information network embedding," in *Proc. of the 24th Int. Conf. on World Wide Web*, Florence, UK, pp. 1067–1077, 2015.

[34] Z. Chen, C. Zhang, Z. Zhao, C. Yao and D. Cai, "Question retrieval for community-based question answering via heterogeneous social influential network," *Neuro-computing*, vol. 285, no. 2, pp. 117–124, 2018.

[35] J. Sun, B. Bandyopadhyay, A. Bashizade, J. Liang, P. Sadayappan *et al.,* "Atp: Directed graph embedding with asymmetric transitivity preservation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 265–272, 2019.

[36] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," arXiv Preprint arXiv: 1301. 3781, 2013.

[37] Y. Dong, Z. Hu, K. Wang, Y. Sun and J. Tang, "Heterogeneous network representation learning," in *Proc. 29th Proc. of the Twenty-Ninth Int. Joint Conf. on Artificial Intelligence*, Yokohama, Japan, pp. 4861–4867, 2020.

[38] J. Tang, M. Qu and Q. Mei, "Pte: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, NSW, Australia, pp. 1165–1174, 2015.

[39] Y. Dong, N. V. Chawla and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. of the 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, NS, Canada, pp. 135–144, 2017.

[40] T. Y. Fu, W. C. Lee and Z. Lei, "HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning," in *CIKM '17: Proc. of the 2017 ACM on Conf. on Information and Knowledge Management*, Singapore, pp. 1–12, 2017.

[41] Y. He, Y. Song, J. Li, C. Ji, J. Peng *et al.,* “Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding,” in *Proc. of the 28th ACM Int. Conf. on Information and Knowledge Management*, Beijing, China, pp. 639–648, 2019.

[42] J. Liu, C. Shi, C. Yang, Z. Lu and P. S. Yu, “A survey on heterogeneous information network based recommender systems: Concepts, methods, applications and resources,” *AI Open*, vol. 3, no. 9, pp. 40–57, 2022.

[43] C. Luo, W. Pang and Z. Wang, “Hete-CF: Social-based collaborative filtering recommendation using heterogeneous relations,” in *2014 IEEE Int. Conf. on Data Mining*, Shenzhen, China, pp. 1–14, 2015.

[44] C. Shi, J. Liu, F. Zhuang, S. Y. Philip and B. Wu, “Integrating heterogeneous information via flexible regularization framework for recommendation,” *Knowledge Information Systems*, vol. 49, no. 3, pp. 835–859, 2016.

[45] Y. Shen, W. Rong, Z. Sun, Y. Ouyang and Z. Xiong, “Question/answer matching for CQA system via combining lexical and sequential information,” in *Twenty-Ninth AAAI Conf. on Artificial Intelligence*, Texas, USA18, 2015.

[46] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, “Distributed representations of words and phrases and their compositionality,” *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.

[47] J. Sun, S. Moosavi, R. Ramnath and S. Parthasarathy, “QDEE: Question difficulty and expertise estimation in community question answering sites,” in *Twelfth Int. AAAI Conf. on Web and Social*, Media, California, USA, pp. 1–14, 2018.

[48] A. Azzam, N. Tazi and A. Hossny, “A question routing technique using deep neural network for communities of question answering,” in *Int. Conf. on Database Systems for Advanced Applications*, Texas, USA, pp. 35–49, 2017.