IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 7, 2020, accepted February 15, 2020, date of publication February 18, 2020, date of current version February 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974893

Hierarchical Attentional Factorization Machines for Expert Recommendation in Community Question Answering

WEIZHAO TANG[1,2,3], TUN LU[1,2,3], DONGSHENG LI[4], (Member, IEEE), HANSU GU[5], AND NING GU[1,2,3]

[1]School of Computer Science, Fudan University, Shanghai 200433, China
[2]Shanghai Key Laboratory of Data Science, Fudan University, Shanghai 200433, China
[3]Shanghai Institute of Intelligent Electronics & Systems, Shanghai 200433, China
[4]IBM Research–China, Beijing 100094, China
[5]Microsoft Inc., Seattle, WA 98052, USA

Corresponding author: Tun Lu (lutun@fudan.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61932007 and Grant 61902075.

ABSTRACT The most challenging task of Community Question Answering (CQA) is to provide high-quality answers to users' questions. Currently, a variety of expert recommendation methods have been proposed and greatly improved the effective matching between questions and potential good answerers. However, the performance of existing methods can be adversely affected by many common factors such as data sparsity and noise problem, which cause less precise user modeling. Moreover, existing methods often model user-question interactions through simple ways, failing to capture the multiple scale interactions of question and answerers, which make it difficult to find answerers who are able to provide the best answers. In this paper, we propose an attention-based variant of Factorization Machines (FM) called Hierarchical Attentional Factorization Machines (HaFMRank) for answerer recommendation in CQA, which not only models the interactions between pairs of individual features but emphasizes the roles of crucial features and pairwise interactions. Specifically, we introduce the within-field attention layer to capture the inner structure of features belonging to the same field, while a feature-interaction attention layer is adopted to examine the importance of each pairwise interaction. A pre-training procedure is designed to generate latent FM feature embedding that encode question context and user history into the training process of HaFMRank. The performance of the proposed HaFMRank is evaluated by using real-world datasets of Stack Exchange and experimental results demonstrate that it outperforms several state-of-the-art methods in best answerer recommendation.

INDEX TERMS Community question answering, attention mechanism, factorization machines, expert recommendation.

I. INTRODUCTION

This decade of years has seen the prosperity of numerous online systems that support question answering (Q&A) activities. In particular, crowdsourcing-based Community Question Answering (CQA) forums provide platforms for people to share and obtain knowledge in the form of asking and answering questions. Existing CQA forums can be roughly divided into two categories: open-domain oriented and specific-domain oriented. CQA sites covering all sorts of topics, such as Yahoo! Answers,[1] Baidu Knows,[2] Quora[3] and Zhihu,[4] are useful information sources for common internet users to broaden the scope of knowledge. Meanwhile, Q&A sites related to specific domains, like Stack Exchange networks,[5] are popular communities that gather professionals and amateurs for exchanging ideas on specific issues. Despite the rapid growth in popularity, CQA faces a number of unique

The associate editor coordinating the review of this manuscript and approving it for publication was Guanjun Liu.

[1]https://answers.yahoo.com
[2]https://zhidao.baidu.com
[3]https://www.quora.com
[4] https://www.zhihu.com
[5]https://www.stackexchange.com

challenges [1]. First, a great number of new questions may be raised every day on a Q&A site of considerable scale. In spite of many unsolved questions that already exist, it takes an active users much effort to meet suitable questions to answer. Second, topical expertise limits the scope of questions that a user can answer competently. Answerers may give irrelevant answers that misguide other viewers without enough understanding of the question in some cases. Third, askers may need to wait a long time until a satisfactory answer appears. Previous studies show that many questions on real-world CQA websites cannot be resolved adequately, indicating that the askers do not receive high-quality answers to their questions in time [2]. All the above situations indicate that the most challenging task of CQA is to invite competent experts to provide users' with useful and worthy answers, which are also called best answers.

Currently, a variety of expert recommendation methods, also known as question routing or expert finding, have been proposed including graph-based [19]–[25] and content-based models [26]–[36], which have greatly improved the effective matching between questions and potential good answerers. However, existing methods often suffer from data sparsity problem. For example, to predict whether a user would be able to provide a best answer to a given question, we need to evaluate his/her performance based on the relevancy between the user's historical activities and the considered question features, including question tags and text. The historical answers of a user usually only cover a small portion of tags and text information, indicating that the interaction between users and question features may be very sparse. Under sparse condition, existing models often fail to work reliably because the connection between user and question cannot be well-established. In situations where input variables are converted by one-hot and multi-hot encoding, the data sparsity problem further deteriorates and severely disturbs the modelling. Among many well-designed matrix factorization based recommendation models and applications [3]–[8], Factorization Machines (FM) [9] is one of the most effective approaches that can handle extremely sparse settings, which makes it an applicable alternative for expert recommendation scenario in CQA. Several FM-based neural network models have been proposed in recent years, e.g., FFM [10], DeepFM [11] and AFM [12]. Among these studies, the concept of field is intuitively introduced. A field can be recognized as a group of features belonging to the same category. However, these models are vulnerable to noisy information contained in feature fields such as question tags and question text, where exist multi-non-zero values if converted to sparse vector representation. Meanwhile, previous research [13] has demonstrated that semantic matching which leverages textual information solely could not achieve satisfactory results and question tags play a more important role than other variables for expert recommendation. Therefore, how to filter out uninformative content from textual information and select the most representative tags for prediction is an important key for field modelling of FM.

Furthermore, most of the previous FM-based variants only model pairwise interactions without considering the importance of interacted features. Along with many successful deep learning applications, attention mechanism [18] has proven its promising prospect for recommendation task by virtue of its capability of allowing the model to place more weights on important features as needed. Attentional Factorization Machines (AFM) [12] is one of the most popular attentional variants of FM, however, it only models the weight of feature interactions without considering the importance of individual features. In CQA, there exist cases that some features in a field are more significant than others. As an example, a tag less similar to the others might impact more on the decision of whether a user would answer a question. What's more, the interaction between features might also contribute differently to the expertise prediction. For example, an answerer might intend to pay different attention to different parts of a question when a question shows up on his/her timeline. Some users may think more of the question body, while others pay more attention to the tags. Thus, it is important to capture the multiple scale interactions of question and answerers by multi-layer attentions.

More concretely, this paper proposes the Hierarchical Attentional Factorization Machines (HaFMRank) for best answerer recommendation in CQA. HaFMRank is a variant of FM and AFM that are enhanced with two levels of attention layers: within-field level and feature-interaction level attention. Within-field attention is designed to assign attentional weights to each individual feature, while feature-interaction attention is able to represent the importance of each pairwise interaction. To fully utilize the external knowledge of a specific CQA site, we also implement a feature embedding that encode user history and corpus context into the training process of HaFMRank. To summarize, this paper has the following contributions:

- We propose a novel network called HaFMRank to facilitate best answerer recommendation in CQA by modeling field-aware feature interaction with factorization machines.
- We implement hierarchical attention layers, i.e., within-field and feature-interaction attentions, to capture the multi-scale complex interaction between features.
- To incorporate external knowledge into the training process of HaFMRank, we apply Word2Vec and TextRank algorithm to encode user history and CQA context and pre-train context-aware latent feature embeddings, aiming to improve the overall performance of HaFMRank.
- We conducted extensive experiments on two real-world datasets and show that HaFMRank significantly outperforms several state-of-the-art CQA expert recommendation models.

The remainder of this paper is structured as follows. Section II presents the related work. Section III illustrates the proposed framework that combines the pre-training of semantic model and HaFMRank model. Section IV

reports the experimental results and we draw the conclusion in Section V.

II. RELATED WORK

In this section, we give a brief review on the related work of expert recommendation in CQA and attention-based recommendation methods. With the rapid development of CQA, all kinds of research issues have been raised to promote the prosperity of Q&A services. In early studies, Harper *et al.* [14] have categorized the then Q&A sites. They further analyzed the predictors of answer quality and suggested that financial incentive and free limits to answerers contribute to the success of a Q&A site. Wang *et al.* [15] found that social relationship help produce more and higher quality of answers. Li *et al.* [16], [17] examined the data of editing behavior extracted from Stack Overflow. Their analysis indicated that the benefits of collaborative editing outweigh its negative effect. Inspired by these pilot studies that focus on CQA, A growing body of expert recommendation researches began to emerge.

The existing expert recommendation work can be divided as two categories: graph-based and content-based approaches. The graph-based methods are built upon the link analysis of the relation between users. Zhang *et al.* [19] proposed a PageRank-like algorithm based on the assumption of the transitivity of expertise to recommend suitable answerers. Similarly, Yang *et al.* [20] applied social network analysis to construct a prestige graph of tasks and users. The model infers relative expertise of users with the use of PageRank algorithm. Liu *et al.* [21] proposed to estimate users' expertise by modelling the pairwise comparison based competition among CQA entities such as non-best answerers, and best answerers. Zhao *et al.* [22] focused on the issue of discovering cold-start experts who only answered a small number of questions. GRLM was proposed to make use of the user-to-user graph to tackle the data sparsity problem if there are many missing values in cold-start users, where common interests and preferences play a role of connecting users upon their interests. Zhu *et al.* [23] infer the category relevance based on similarity and rank user authority in extended category graph. Liu *et al.* [24] tackled the sparsity problem by integrating topic representations from CQA data with network structure from the viewpoint of knowledge graph embedding. Sun *et al.* [25] proposed QDEE that applies Expertise Gain Assumption (EGA) to alleviate data sparsity problem and constructed competition graphs from the users' historical activities. QDEE interprets the hierarchical structure of competition graphs as the question difficulty and user expertise.

In terms of content-based approaches, researchers usually consider expert recommendation as a topic modelling problem. Liu *et al.* [26] devised the CQARank model that takes both user latent topic and expertise estimation into consideration, aiming to find experts with both similar topical preference and high-level topical expertise. Zhao *et al.* [27] proposed a topic-level expert learning framework which simultaneously provides the topic of questions and identifies experts on each topic. Mumtaz *et al.* [28] applied the state-of-the-art embedding word technique to capture domain-specific semantics that matches user expertise with question content. Bouguessa *et al.* [29] proposed a method to solve the problem of determining how many users should be selected as experts from a user list ranked by number of best answers. They also argued that best answer is important to estimate user expertise score. Later, they proposed another probabilistic model (BMM) to detect authoritative users in [30] and Sahu *et al.* [31] made further efforts to demonstrate the effectiveness of BMM for topic expert identification. Hanrahan *et al.* [32] proposed several feasible indicators to represent the difficulty of questions, including the duration between the time when the question was asked and the time when an answer was marked as the best answer. Huna *et al.* [33] adopted the assumption proposed in [32] and calculated the expertise of a user with accentuation on the difficulty of relevant questions users have answered. A user gains greater reputation for asking difficult and useful questions and for providing useful answers on other difficult questions. Yang *et al.* [34] described an expertise metric called ''Mean Expertise Contribution'' that takes debatableness and utility into account. Debatableness is related to the number of answers to a question, while utility is negatively correlated to the ranking of an answer among all the answers in a given question. Sun *et al.* [13] handles cold-start problem of question routing in CQA by introduction factorization machines. Their results indicate that critical features such as question tags play a more important role than other content. Unlike previous approaches, given that social relations between two users provides a strong evidence for them to have common background, Zhao *et al.* [35] proposed RMNL to leverage social relations and triplet constraints to tackle question answering problems in CQAs. They considered users' online social network information as key feature to facilitate question routing work.

Considering the current progress in deep learning, attention mechanism [18] has been widely applied as an effective technique in various fields of deep learning from natural language processing to computer vision. By virtue of its capability to focus and place more weight on the relevant parts of input as needed, attention mechanism has shown its promising prospect in recommendation task. Xiao *et al.* [12] proposed a novel recommender framework (AFM) that first combines attention mechanism with factorization machines and enables feature interaction contribute differently according to the importance. Chen *et al.* [37] developed an attention-based model that captures category-wise user intention and facilitates conventional FM model for top-k recommendation. Zhu *et al.* [38] addressed the issue of personalized new recommendation with a deep attention neural network that is capable of considering the user's history sequential data along with current preference. Wu *et al.* [39] present a hierarchical attention network for social contextual image recommendation. The model learns from heterogeneous data sources

and attend differently to more or less important content. Zhou *et al.* [40] utilized a self-attention based sequential framework to project user representation into multiple latent spaces and models user behavior for personalized recommendation. Wu *et al.* [41] facilitate news recommendation by modeling the contextual interaction between words and news with multi-head self-attention. Cong *et al.* [42] distinguish the importance of reviews at both word level and sentence level using a hierarchical attention-based network for e-commerce recommendation. Similarly, Wang *et al.* [43] proposed a two-level attention mechanism with time encoding and achieved good performance in the next-item recommendation.

Compared with the previous studies, the proposed framework mainly differs in the following aspects. 1) We first address the problem of expert recommendation in CQA by extending factorization machines with attention mechanism. The proposed approach is expected to solve the issue of data sparsity and noise, which is very common in the scenario of CQA. 2) External and context-related knowledge is incorporated into the training process of FM-based method, aiming to improve the overall recommendation performance. As demonstrated in the experiments, our framework outperforms the other expert recommendation approaches in three evaluation metrics.

III. THE PROPOSED APPROACH

A. PROBLEM STATEMENT

Given a question Q, the objective of the expert recommendation task is to route Q to experts who are most likely to provide high-quality answers. A user can be considered as an expert to answer Q if he/she has already answered a set of related questions from which we can infer his/her expertise to answer the question. More specifically, if we are given a newly posed question Q and n number of users $U = \{u_1, u_2, \ldots, u_n\}$ having a set of m answer posts $P = \{p_1, p_2, \ldots, p_m\}$, our goal is to find a ranked list of k users $A = \{a_1, a_2, \ldots, a_k\} \in U$ who exhibit both domain expertise and interest of answering Q. We then recommend best answerers from list A according to the requirement. We consider the users' past activities as an indicator of their expertise and interest on a question and estimate their performance through expert recommendation algorithms. As we will explain in the following sections, our work leverages users' contribution (i.e., answer records) to match questions and experts. Therefore, a key challenge we face is to learn a mapping function $f(Q, u) \rightarrow \mathbb{R}$ which computes the ranking score for answerers that indicates a user's expertise and willingness for matching the newly posted question. The answerer with highest ranking score would be selected as the predicted provider of the best answer. Table 1 lists the frequently used notations and descriptions in this paper.

TABLE 1. Notations and description.

Notation	Description
y	vectors of the targets
$\hat{y}$	prediction of the models
$\mathbf{X}$	feature vectors, where x_i represents the i-th vector
k	the dimension of the latent embedding vector
d, t	attention factors, the hidden layer size of the attention network
Θ	model parameters
α_i^{fi}	within-field attention score of i-th feature in field f
a_{ij}	feature-interaction attention score of the inner product between the i-th and j-th feature embeddings
λ	regularization strength

B. OVERALL FRAMEWORK

This subsection mainly focuses on the description of the framework of the proposed CQA expert recommendation. We first pre-trains semantic representations of users and questions from CQA context with the use of word embedding and keyword extraction techniques, and then completes expert recommendation task based on the newly developed attention-based FM model. The overall framework of our proposed approach mainly consists of two functional modules: pre-training procedure for the latent representation of input features and Hierarchical Attentional Factorization Machine (HaFMRank) algorithm. HaFMRank can be described as a multi-layered neural network that models the attention-aware representation of multi-valent field features while assigning attentional weights to the pairwise interactions of features. Fig. 1 illustrates the overall workflow of the proposed framework. As presented, we consider three feature fields, i.e. AnswererID, Question Tags and Question Text as the components of each input. Specifically, each AnswererID is converted into embedding vector by one-hot encoding. Question Tags, which usually contains several different terms, could be transformed into multi-valent embedding vector by multi-hot encoding. In terms of the multi-hot representation of Question Text, we create a bag of words (BoW) model that consists of the most frequently appearing terms in a CQA site and represent the corresponding text with this BoW model.

As mentioned above, an answer record used by our method usually consists of features from multiple fields (e.g., answerer, question tags, question text) that are converted into high-dimensional sparse vectors by one-hot or multi-hot encoding. The embedding layer of proposed algorithm will embed these high-dimensional sparse vectors into dense representation in order to learn the strength of interaction between features. The interaction of features is modelled by the latent vectors. In general applications of factorization machines, the input of latent vectors are usually initialized with random values, which do not relate features with its context knowledge.

Differing from traditional FM application such as Click-Through Rate (CTR) prediction, the features used in

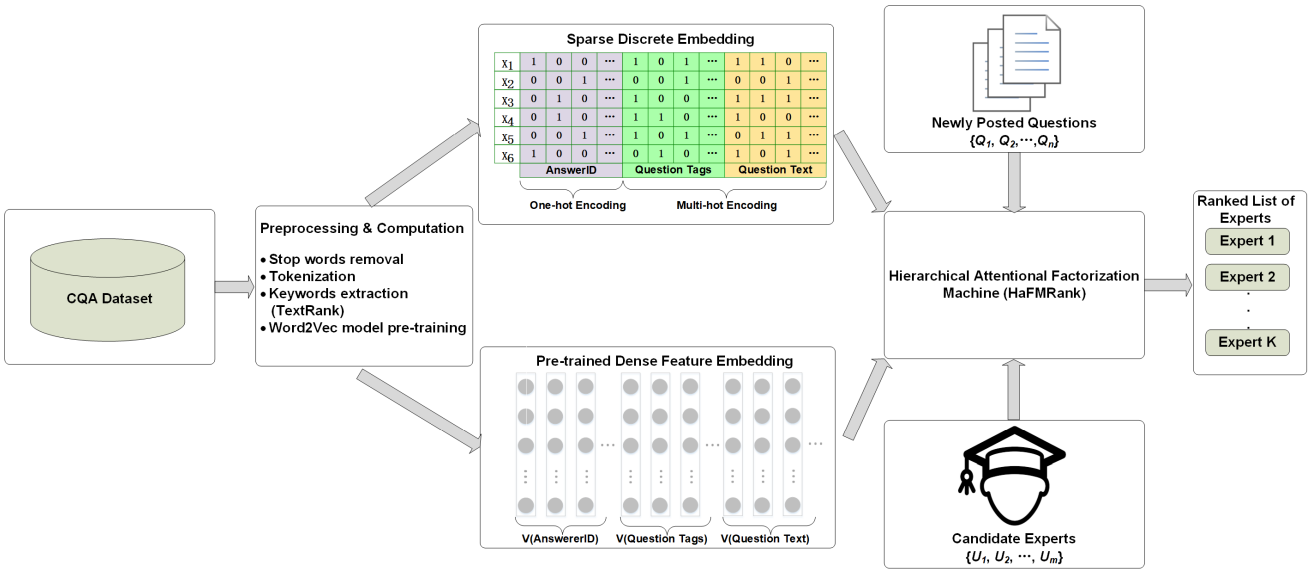


FIGURE 1. The overall workflow of the proposed framework.

expert recommendation can also be effectively represented by its external semantic meaning. Thus, we study the incorporation of external, context-related knowledge in the training process of factorization machines for the best answerer recommendation in CQA. As an example, feeding similarity knowledge between given tags and user expertise into FM may benefit the modelling of feature interactions. We apply word embedding technique and keyword extraction method to build initialization process of the input latent embeddings for our approach. Semantic vector representation in CQA can be computed in various ways. One popular technique is the vector space model [44], which considers users' posts as a bag of words and uses occurrences count of words as features. We first learn a vector representation of words from CQA corpus using Word2Vec model [45]. The Word2Vec model uses a neural network to learn distributed word embeddings. The quintessential outputs of a Word2Vec model can be logically used as the dense embedding inputs for FM-based models. This enables the recommendation algorithm to take associated context from a CQA site into account when generating the final models. Thus, we can train this model that deliver domain-specific word embeddings on the dataset for each word in the corpora, including tags and other frequently appeared words.

To pre-train an expertise representation embedding for the field of AnswererID. We apply a keyword extraction procedure that obtains a set of keywords $\{kw_1, kw_2, \ldots, kw_k\}$ from the question and answer posts of users. We conclude the history of a user as his/her answers, answered questions and raised questions. We combine all relevant tags, question text and answers as the user-specific corpus. In order to extract the most salient words from the corpus, we apply TextRank [46] algorithm to obtain the top-N keywords from the corpus for expertise representation. Each set of keywords is then used to represent each individual user in an embedding space. We then generate each user-related expertise embedding. To do so, we make use of vectors averaging embedding. Vectors averaging is an intuitive yet powerful approach for text representation through embedding. In this work, we compute the average of the word vectors of each keyword $kw_i \in K_u$ in the user's corpus to obtain the semantic representation of users' expertise. That is, for each user u, we compute:

$$V_u = \frac{1}{k}\sum_{i=1}^{k} WE(\mathrm{kw}_{ui}) \tag{1}$$

where Vu is the vector representation of user u, k is the number of top keywords obtained from the user-related corpus and $WE(kw_{ui})$ is the word embedding of kw_{ui} trained by Word2Vector model. The embedding representation for tags and question texts, including title and body, can be conveniently represented by the word embedding pre-trained in the above steps. Specifically, the latent embedding of each top-k keyword in question title and body can be initialized with the aforementioned Word2Vector model.

C. HIERARCHICAL ATTENTIONAL FACTORIZATION MACHINE

As one of the state-of-the-art recommendation algorithms, factorization machines (FM) [9] models all interactions between pairs of non-zero values in the embedding vector using factorized interaction parameters. Given a sparse input vector $\boldsymbol{x} \in \mathbb{R}^p$, where p denotes the dimension of feature space, a 2-order FM model can be described as follows:

$$\hat{y}_{FM}(\boldsymbol{x}) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p}\sum_{j=i+1}^{p} < \mathbf{v_i}, \mathbf{v_j} > x_i x_j \tag{2}$$

where $w_0 \in \mathbb{R}$ is the global bias, $w_i \in \mathbb{R}$ denotes the strength of the i-th features. The interaction between i-th and j-th features is modelled by the inner product of the latent vectors, i.e. $< \mathbf{v}_i, \mathbf{v}_j >= \sum_{f=1}^{k} v_{i,f} v_{j,f}$, where k represents the dimensionality of the latent vectors. In practice, factorization machines is supposed to learn the relationship about different features under sparse inputs and predict target scores for tasks including regression, classification and ranking.

Algorithm 1 Hierachical Attentional Factorization Machines

Input: Training dataset **X,** pre-trained latent embedding matrix **V**, attention factors t and d, regularization A
Output: Model Parameters Θ

initialize model parameters except V with random values from uniform distribution $(-1, 1)$
repeat
  shuffle Dataset **X**
  for $\mathbf{x} \in \mathbf{X}$ **do**
    set p as the dimension of x
    for each field **/ do**
      for $x_i \in \mathbf{x}_f$ **do**
        compute each f_{att} according to Eq. (4)
        compute α_i^f according to Eq. (5)
      end for
    end for
    for $i - 1$ to p **do**
      for $j - i + 1$ to p **do**
        compute $a'_{i,j}$ and $a_{i,j}$ according to Eq. (8) and Eq. (9) respectively
      end for
    end for
    compute $\hat{y}(x)$ according to Eq. (10)
    update Θ with gradient descent.
  end for
until convergence
return Θ

For the sake of capturing the importance of each feature and feature interaction, we propose an attention-based factorization machine models, i.e. Hierarchical Attentional Factorization Machine (HaFMRank). HaFMRank takes a sparse vector of features as input. Each feature can be categorized into one field. The embedding layer embeds each non-zero feature into a dense vector. It is worth noting that most FM variants assume that each field only contains one non-zero feature. The within-field attention component focuses on capturing inner structure of each field that contains multi-non-zero values. The embedding layer can model different types of features such as categorical and textual features, which means the proposed model is able to process these features together. The within-field attention mechanism assigns different attentional weights to the feature embedding vectors in each field to obtain a set of weighted embeddings. The upper layer of attention models the interactions between the weighted and unweighted embeddings of each individual non-zero feature to get the pairwise interactions. Pairwise-level attention is applied to generate the attentional scores for different feature interactions to obtain the attention-weighted interactions which can be used for the final prediction.

To implement HaFMRank, we first generate the pairwise interaction layer as proposed in [12], which models interactions between different features. It expands m vectors to $m(m\text{-}1)/2$ interacted vectors, where each interacted vector is the Hadamard product of two different vectors to encode their interaction. Specifically, let the set of non-zero features in the feature vector be $\boldsymbol{x}$, and the output of the embedding layer be $\varepsilon = \{\mathbf{v}_i x_i\}_{i \in X}$. We can then represent the output of the pair-wise interaction layer as the summation of a set of interacted vectors:

$$f_{\mathrm{PI}}(\varepsilon) = \sum_{i=1}^{p} \sum_{j=i+1}^{p} (\mathbf{v}_i \odot \mathbf{v}_j) x_i x_j \tag{3}$$

where $\odot$ denotes the Hadamard (element-wise) product, p is the dimension of input features.

1) WITHIN-FIELD ATTENTION (WFA) LAYER

Attention mechanism has been introduced to neural network modelling and widely used in many deep learning tasks. The aforementioned pair-wise interaction layer models the features in a specific field with the same weights. However, in a multi-valent feature field, not all non-zero features contribute equally to the representation of the input. Uninformative features can be sometimes considered as noise which would result in poor performance. For example, there could exist several terms in the tag field of each input. But we have to pay more attention to the most significant tags and try to ignore those less informative ones. The goal of within-field attention mechanism is to assign features in the same field with attentive weights to represent the field. The rationale we evaluate the importance of features is based on the similarity between terms. Thus, we propose the computation of attention score as follows:

$$f_{att}(\mathbf{v}_i^f, \mathbf{v}_j^f) = \mathbf{h}_f^T \, tanh(\mathbf{W}_{f1} \mathbf{v}_i^f + \mathbf{W}_{f2} \mathbf{v}_j^f + \mathbf{b}_f) \tag{4}$$

$$\begin{aligned} \alpha_i^f &= \mathrm{softmax}(f_{att}(v_i^f, v_i^f)) \\ &= \frac{\exp(f_{att}(v_i^f, v_i^f))}{\sum_{\exp(f_{att}(v_i^f, v_j^f))}} \end{aligned} \tag{5}$$

where α_i^f can be viewed as the importance of i-th feature in its field f, $\mathbf{W}_{f1}, \mathbf{W}_{f2} \in \mathbb{R}^{d \times k}$, $\mathbf{b}_f \in \mathbb{R}^d$, and $\mathbf{h}_f \in \mathbb{R}^d$ are parameters of the within-field attention network. And d denotes the hidden layer size of the attention network, i.e. attention factor [12]. The attentional weights are normalized through the softmax function. We can then represent the pair-wise interaction layer by the summation of the Hadamard product of attention-score weighted vectors:

$$f_{PI}(\varepsilon) = \sum_{i=1}^{p} \sum_{j=i+1}^{p} (\alpha_i^{f_i} \mathbf{v}_i \odot \alpha_i^{f_j} \mathbf{v}_j) x_i x_j \tag{6}$$

2) FEATURE INTERACTION ATTENTION (FIA) LAYER

Although we have assigned within-field attentional weight to each individual feature, the weight of each pairwise interaction still remains 1, which might probably undermine the generalization capability of the FM model. Thus, we investigate the application of attention mechanism on the pairwise feature interactions (i.e. feature interaction attention) by placing attentional score on the Hadamard product of feature embeddings. The design is to allow different feature interactions contribute differently when aggregating the inner products into a single representation. Inspired by AFM model, we propose to employ the attention mechanism on within-field attention-weighted feature interactions by performing a weighted sum on the interacted vectors:

$$f_{Att}(f_{\text{PI}}(\varepsilon)) = \sum_{i=1}^{p} \sum_{j=i+1}^{p} a_{ij}(\alpha_i^{f_i}\ \mathbf{v}_i \odot \alpha_j^{f_j}\ \mathbf{v}_j)x_i x_j \tag{7}$$

where a_{ij} is the attention score that can be interpreted as the contribution made by the interaction between i-th and j-th features. We use the multi-layered perceptron (MLP) based attention network to estimate the attention score of each feature interaction and take the within-field attention-weighted embedding $\alpha_i^{f_i}\mathbf{v}_i$ from each field as input:

$$a'_{ij} = \mathbf{h}_{FI}^T \text{relu}(\mathbf{W}_{FI}(\alpha_i^{f_i} \quad \mathbf{v}_i \odot \alpha_j^{f_j}\mathbf{v}_j) + \mathbf{b}_{FI}) \tag{8}$$

$$a_{ij} = \text{softmax}(a'_{ij}) = \frac{\exp(a'_{ij})}{\sum_{m=1}^{p}\sum_{n=m+1}^{p}\exp(a'_{mn})} \tag{9}$$

where $\mathbf{W}_{FI} \in \mathbb{R}^{t\times k}, \mathbf{b}_{FI} \in \mathbb{R}^t$, and $\mathbf{h}_{FI} \in \mathbb{R}^t$ are the parameters of the pairwise interaction level attention network, specifically, t stands for the hidden layer size of the attention pooling-based network. Similar to the WFA layer, we obtain the pairwise interaction attentional weights by normalizing the initial FIA weight a_{ij} through the softmax function. The output of the sum pooling layer with within-field and pairwise interaction level attentions is a k dimensional vector, which compresses all interactions in the embedding space. Based on the attentional features in multi-valent field that contains answer entry information, we propose the final target score of HaFMRank as follows:

$$\hat{y}_{HA\_FM}(\mathbf{x}) = w_0 + \sum_{i=1}^{p} w_i x_i + \mathbf{p}^T \sum_{i=1}^{p}\sum_{j=i+1}^{p} a_{ij}(\alpha_i^{f_i}\mathbf{v}_i \odot \alpha_i^{f_j}\mathbf{v}_j)x_i x_j \tag{10}$$

where $\mathbf{p} \in \mathbb{R}^k$ denotes the weights for the prediction layer. For simplicity, we fix $\mathbf{p}$ to $\mathbf{1}$ in this study. Finally, we write the set of all the model parameters compactly as:

$$\Theta = \{w_0, \{w_i\}_{i=1}^p, \{\mathbf{v}_i\}_{i=1}^p, \{\mathbf{W}_{f1}, \mathbf{W}_{f2}, \mathbf{h}_f, \mathbf{b}_f\}_{f\in\mathbf{F}}, \mathbf{W}_{FI}, \mathbf{h}_{FI}, \mathbf{b}_{FI}\}$$

D. OBJECTIVE FUNCTION AND OPTIMIZATION

To minimize the error between the predicted value $\hat{y}$ and the actual value y, we need to obtain the optimal parameters Θ by the training of HaFMRank. For regression task where the target $y(x)$ is a real value, a common objective function is the squared loss:

$$Loss = \sum_{\mathbf{x}\in D} (\hat{y}_{HaFMRank}(\mathbf{x}) - y(\mathbf{x}))^2 \tag{11}$$

To prevent the possible over-fitting we apply L_2 regularization on the linear weight matrix $\mathbf{W}$. Hence, the actual objective function we optimize is:

$$Loss = \sum_{\mathbf{x}\in D} (\hat{y}_{HaFMRank}(\mathbf{x}) - y(\mathbf{x}))^2 + \lambda \parallel \mathbf{W} \parallel^2 \tag{12}$$

To optimize the objective function, we utilize Stochastic Gradient Descent (SGD) algorithm to learn the model parameters. For the sake of predicting the topical interest and expertise of a user, we set the real value of y as follows:

$$y = Normalize(\sum_{i=1}^{T} number\_of\_answer(tag_{i,u}) + upvotes) \tag{13}$$

where y consists of two parts: topical interest and topical expertise. The topical interest of user u to question q is represented by the total number of questions answered by u that contain q's tags, i.e. $number\_of\_answer(tag_{i,u})$, while topical expertise of user u to question q is represented by the received upvotes. The final target value is rescaled by the min-max normalization function. The detailed learning algorithm is presented in Algorithm 1.

IV. EXPERIMENTS

A. DATASET DESCRIPTION

Stack Exchange (SE) is a large-scale online question-answering community network which consists of more than 170 Q&A sub-sites. Each sub-site of SE concentrates on a specific knowledge domain. Amongst these sites, Stack Overflow[6] is one of the most popular online sites for software development issues. We downloaded the public dataset of Stack Exchange[7] since its launch in August 2008 to December 2016. The dataset consists of more than 40 million questions and answers posted by millions of users. The data dump of Stack Exchange contains all registered users' activity logs, including questions, answers, votes, edits and comments.

Similar to [36] and [47], we select users as active users who asked and answered more than 100 times in the first half of 2015 from Stack Overflow. We extracted all questions and answer records posted by the active users in the latter 6 months of 2015 as their history, on which we pre-trained the latent embedding and created training set. Then we selected questions answered by no less than 2 active users along with their answers in the first half of 2016 as testing instances. To test the performance on testing questions,

[6]https://stackoverflow.com
[7]https://archive.org/download/stackexchange

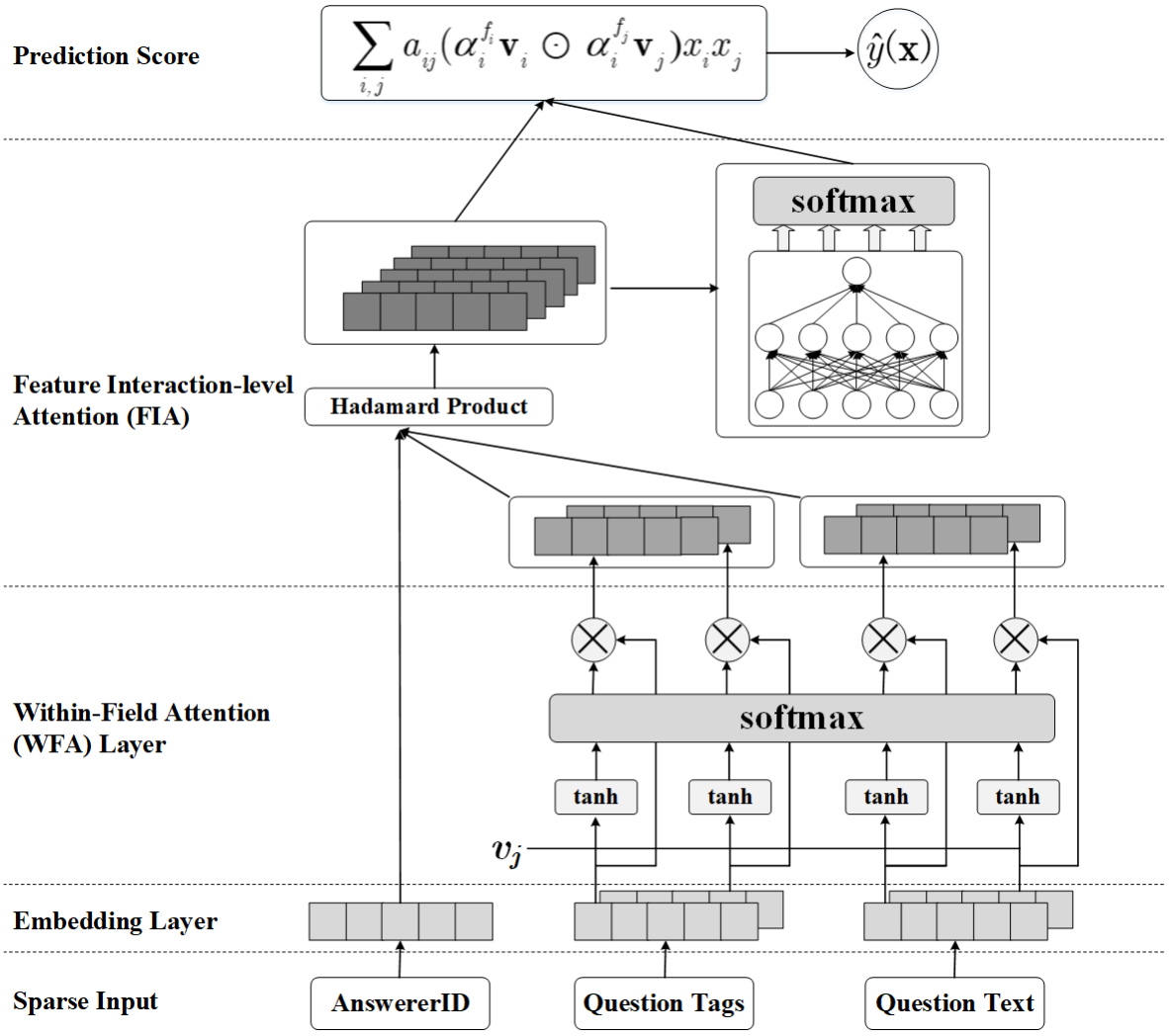


FIGURE 2. The network architecture of HaFMRank model.

we created a candidate answerer set of 15 answerers that includes all answerers of testing question who are active users and some other users randomly selected from the active users. We choose the answerer with the highest predicted ranking score as the recommendation and the users who received highest votes are the ground truth. In the same way, we created another dataset from Math community.[8] The overview of the dataset is illustrated in Table 2.

TABLE 2. Overview of dataset.

Stack Overflow	Training set	Active Users	1940
		Questions	225363
		Answers	251359
	Test set	Questions	14450
Math	Training set	Active Users	895
		Questions	57933
		Answers	88200
	Test set	Questions	9969

B. EVALUATION METRICS

To measure answering possibility, we use popular infor-mation retrieval metrics Precision@K and MRR. Besides, a user could also be ranked via the metric Hit@K for finding the best answerers. Relevance labels 1 and 0 are adopted to calculate these metrics, which indicate whether or not a given user is the best answerer of a question.

Precision@K measures the number of recommended an-swerers who are the real suitable answerers in top-K of the predicted ranking of all users, normalized by the cutoff value K and averaged over all questions. Actually, if we only consider the best answers as ground truth, we evaluate pre-cisions at cut-off levels K = 1, i.e. Precision@1, which is defined as:

$$Precision@1 = \frac{\{q \in Q | Rank_{i,best} = 1\}}{|Q|} \quad (14)$$

where |Q | is the number of questions and $Rank_{i,Best}$ is the rank of the best answerer of question Q_i returned by the recommendation algorithm.

[8]https://math.stackexchange.com

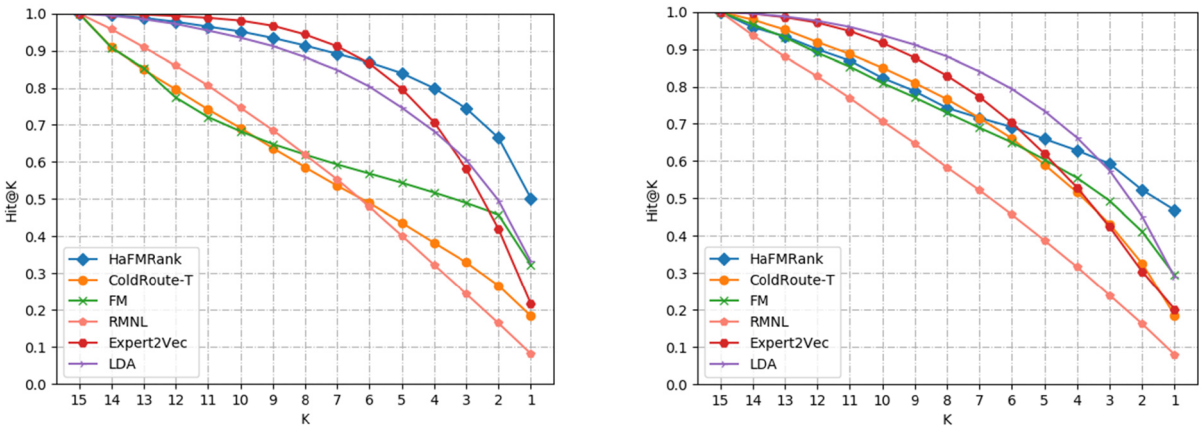


FIGURE 3. Performance comparison of methods on Hit@K (a) Stack Overflow (b) Math.

Mean Reciprocal Rank (MRR) is the average of the Reciprocal Rank (RR) for a set of questions. The mean reciprocal rank for a given set of questions Q is defined as:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{Rank_{i,best}} \tag{15}$$

Hit@K is applied to measure the average number of times that the best answerer (ground truth) can be found in the top-k predicted answerers. It is noteworthy that Hit@1 is equal to Precision@1. The calculation formula of Hit@K is defined as:

$$Hit@k = \frac{\{q \in Q | Rank_{q,best} <= k\}}{|Q|} \tag{16}$$

C. BASELINE METHODS

To evaluate the effectiveness of our framework, we compare against some previous related works including probabilistic topic models, FM-based techniques and semantic matching methods as follows:

1) ColdRoute [13]

ColdRoute is a FM-based expert recommendation approach in CQA, which investigates the roles of question tags, asker and question body through FM model. Among the many variant of ColdRoute compared in the paper, we implemented ColdRoute-T that explore the importance of question tags by using triples of <QuestionID, AnswererID, Tags> due to its outperformance over the other variants.

2) RMNL [35]

RMNL was proposed to leverage social relations and triplet constraints to tackle question answering problems in CQAs. RMNL used users' social network follower/followee information to enhance experts finding ability. Please note that since Stack Exchange does not provide the common following feature as other popular SNS, we consider whether a user has marked another user's answer as accepted answers as an indicator of strong relationship.

3) Expert2Vec [28]

Expert2Vec is a question routing framework that makes use of textual content as evidences of user expertise within the community and also leverages the community feedback and interactions of users with one another to better match user expertise for routing questions.

In addition, we also compare our method with classical approaches including FM and LDA. We use LDA to extract the topic vector from user's history and compute the similarity between user-related topic vector and the given question. The number of LDA topics is set as 100. The embedding size of the latent vectors of FM-based models (i.e. HaFMRank, ColdRoute-T and FM) is 100, same as the size of the word embedding used in Expert2Vec. The hidden layer sizes (i.e. attention factor) of both attention levels in HaFMRank are set at 20 and we will report the influence of attention factor in later section.

D. RESULTS AND DISCUSSION

1) PERFORMANCE COMPARISON

In this subsection, we compare the performance of the proposed HaFMRank question routing framework and the baselines with respect to three evaluation metrics, i.e., Precision@1, MRR and Hit@K. Table 3 shows the overall performance of all the recommendation methods on Stack Overflow and Math. The best values are indicated in bold print. Fig. 3 shows the performance of all methods under metric of Hit@K from Top-15 to Top-1 in both datasets. From the table and figure, we can observe that:

- The proposed HaFMRank consistently achieves the best performance in all cases under the metrics of Precision@1 and MRR on both Stack Overflow and Math datasets. Specifically, HaFMRank achieves the improvements of 17.0% and 13.8% in terms of Precision@1 and MRR compared with the second best methods for Stack Overflow. For Math, the improvements are 17.5% and 8.7% in terms of Precision@1 and MRR. The advantage demonstrates the effectiveness of the proposed

TABLE 3. Overall Performance comparison between HaFMRank and the other baselines.

Dataset	Method	Prec@1/Hit@1	MRR	Hit@3	Hit@5
Stack Overflow	HaFMRank	**0.502**	**0.651**	**0.745**	**0.840**
	FM	0.324	0.455	0.490	0.544
	ColdRoute-T	0.184	0.327	0.329	0.436
	RMNL	0.082	0.250	0.244	0.402
	Expert2Vec	0.217	0.420	0.581	0.793
	LDA	0.332	0.513	0.606	0.746
Math	HaFMRank	**0.469**	**0.568**	**0.593**	0.659
	FM	0.294	0.447	0.494	0.604
	ColdRoute-T	0.186	0.374	0.432	0.591
	RMNL	0.079	0.245	0.239	0.388
	Expert2Vec	0.202	0.365	0.424	0.619
	LDA	0.289	0.481	0.574	**0.734**

HaFMRank. It indicates that our framework is more powerful than the other baselines for the task of recommending best answerer in CQA.

- HaFMRank outperforms ColdRoute-T and FM under all measurements with a considerable amount of margin. More specifically, the average improvements by HaFMRank over FM are 17.6% and 20.8% at Precision@1 and MRR. It demonstrates that introducing attention mechanism will promote the capability of the FM-based model in our task. In addition, the performance of FM is better than ColdRoute-T. The prime difference between these two methods lies in the selection of the target values. We consider both user's answer count and received votes for training FM model, while ColdRoute-T only focuses on the latter. We can argue that it is important to jointly model user's topical interest and expertise in expert recommendation work.
- As shown in Figure 3, semantic matching based models (i.e. Expert2Vec and LDA) outperforms the other methods including HaFMRank in over half of all cases under Hit@K. In Stack Overflow, Expert2Vec performs best under Hit@K from Top-15 to Top-6, while LDA achieves the best result from Top-15 to Top-4 in Math. However, HaFMRank surpasses Expert2Vec and LDA when K decreases to a small number. This indicates that semantic matching based models tend to rank most ground-truth answerers in the top half recommendations, but HaRMRank is still the best recommendation algorithm when only a small number of answerers should be invited.
- Note that RMNL performs worst in almost all cases, this is probably caused by its requirement of social network relationship. It is widely shared that Stack Exchange is not well-known for its social characteristics and the connection between users is not as strong as that in common SNS such as Facebook and Twitter or other Q&A sites like Quora and Zhihu, which undermines the potential of the use of social relationship for expert recommendation on SE forums.

2) IMPACT OF MODEL PARAMETERS

In this section, we investigate the influence of feature fields and attention factors on our model. As mentioned above, we encode each answer record into a sparse vector that consists of three feature fields, i.e., AnswererID, Question Tags and Question Text. Note that AnswererID is a necessary field when selecting suitable answerers for a given question. Thus, the alternative combinations of feature fields are 'AnswererID + Question Tags' ('AID+Tags'), 'AnswererID + Question Text' (AID +Text), and 'AnswerID+Question Tags+ Question Text' (AID+ Tags+Text). Fig. 4 shows the performance comparison of the three field combinations on HaFMRank. It can be found that the best performance of HaFMRank is achieved with the use of all three fields. i.e. 'AID+Tags+Text'. This result indicates that all the three fields contribute to the prediction of best answerer and the hierarchical attention structure works effectively for filtering out noisy information. It is also noteworthy that 'AID + Tags' performs better than 'AID + Text'. The reason is probably because question tags usually contain concise and accurate information that could be easily captured by prediction models.

We further investigate the impact of attention factor, i.e. the hidden layer size of attention networks. Note that there exist

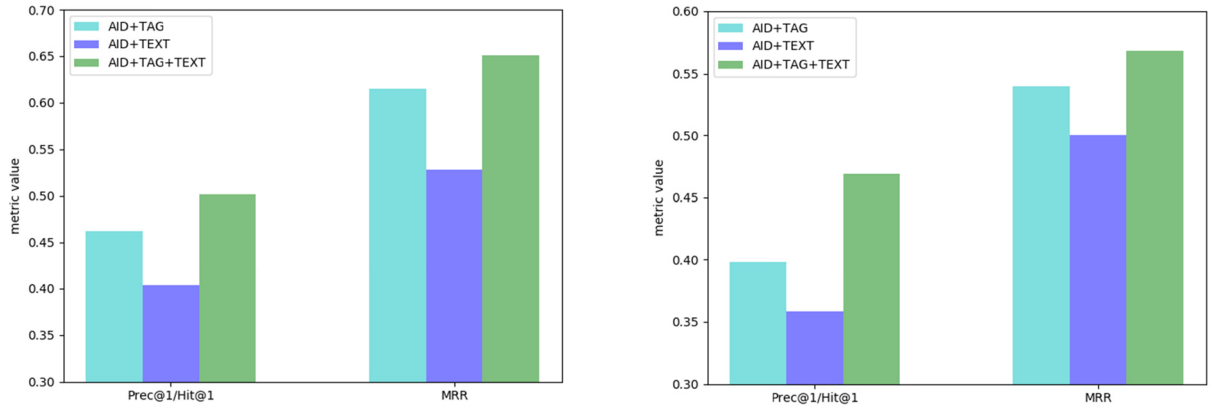


FIGURE 4. Performance comparison w.r.t different combinations of feature fields (a) Stack Overflow (b) Math.

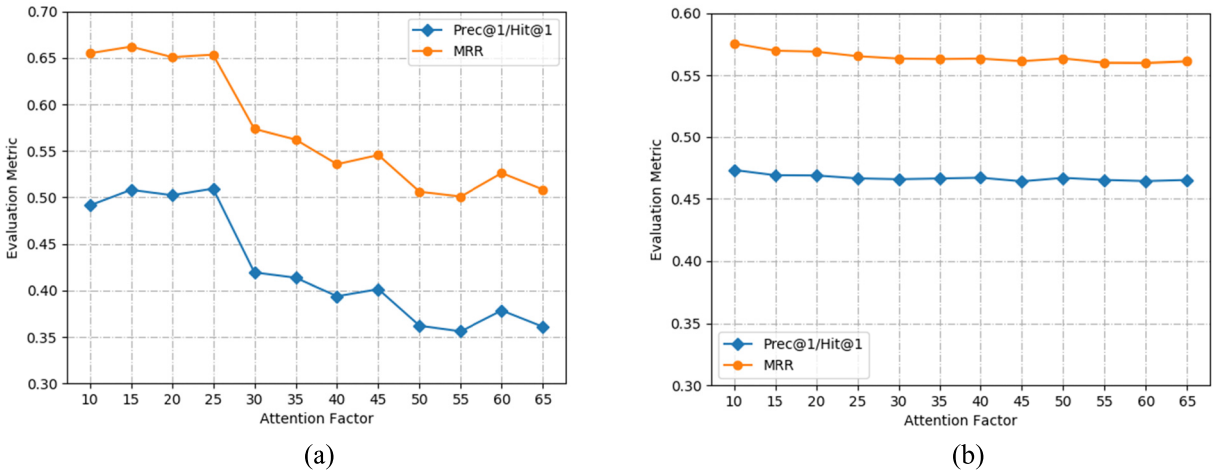


FIGURE 5. Performance comparison w.r.t different attention factors (a) Stack Overflow (b) Math.

two layers of attention network in our model, we bind the two attention factors with each other for simplicity, assuming that they share the same value. Fig. 5 shows the performance in terms of Precision@1 and MRR with regard to different attention factors. It can be observed that high dimension of attention network will not be helpful to build effective feature interaction in the proposed model. For Stack Overflow, increasing the attention factor undermines the performance of HaFMRank if the number exceeds 25. For Math, the performance remains rather stable across the testing range, which means the model is not that responsive to the change of attention factors on this specific dataset. Based on the result, setting attention factors to 20 for both within-field level and feature-interaction level of attention networks is a good trade-off between computation cost and predicting performance.

3) ANALYSIS OF ATTENTION NETWORK STRUCTURE

To evaluate the contribution of each level of attention network in HaFMRank, we conduct experiments to analyze each attention layer. The result is shown in Table 4. We first removed the within-field attention from HaFMRank and denoted the revised model as HaFMRank-WFA. Similarly, HaFMRank-FIA represents the model that only removes feature-interaction level attention network. HaRMRank achieves better performance compared with HaFMRank-WFA and HaFMRank-FIA. This indicates hierarchical attention mechanism is capable of capturing multi-scale complex interaction between features and a single layer of attention network is not enough to construct a robust model for expert recommendation in CQA. HaFMRank-FIA outperforms HaFMRank-WFA by a large margin, which demonstrates that filtering out uninformative individual features plays a more crucial role on this task. Given that even the basic FM model achieves better result than HaFMRank-WFA, modeling feature-interaction level attention without considering the importance of individual feature might lead to negative impact. The reason may be that the influence of noisy features would be enhanced by introducing FIA in some extreme cases. Finally, the proposed two-layer attention-based FM model performs better than the two single attention layer model under two measurements, i.e. Precision@1 and

TABLE 4. Influence of different attention network structure at Precision@1 and MRR.

Evaluation Metrics	Datasets	Attention Network Structure		
		HaFMRank	HaFMRank -WFA	HaFMRank -FIA
Prec@1/Hit@1	Stack Overflow	**0.502**	0.265	0.388
	Math	**0.469**	0.250	0.326
MRR	Stack Overflow	**0.651**	0.402	0.548
	Math	**0.568**	0.428	0.483

MRR. The results demonstrate that effectiveness of the fusion of hierarchical attention networks with factorization machines in the given task.

V. CONCLUSION

In this study, we addressed the issue of performing expert recommendation for newly-posed questions in CQA using a framework combining attention mechanism and factorization machines. After studying the characteristics of CQA sites and the common limitations of previous researches, we present a hierarchical attentional factorization machine model (HaFMRank) along with a pre-training procedure for context-aware feature embeddings, which models attention-weighted feature interactions. Extensive experiments demonstrate the effectiveness of HaFMRank in real-world datasets.

In the future, we would like to examine whether introducing more complex structure of attention network could improve the capability of capturing highly informative features. Besides, we will explore the effect of deep models like CNN and LSTM on processing the textual part of questions and see whether it can further improve the performance. Another promising direction we are interested in is the transfer learning of expertise between different knowledge domains. We believe this is a possible solution for pre-training user expertise representation in a cold-start CQA scenario.

REFERENCES

[1] X. Wang, C. Huang, L. Yao, B. Benatallah, and M. Dong, "A survey on expert recommendation in community question answering," *J. Comput. Sci. Technol.*, vol. 33, no. 4, pp. 625–653, Jul. 2018.

[2] B. Li and I. King, "Routing questions to appropriate answerers in community question answering services," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2010, pp. 1585–1588.

[3] Y. Jia, C. Zhang, Q. Lu, and P. Wang, "Users' brands preference based on SVD++ in recommender systems," in *Proc. IEEE Workshop Adv. Res. Technol. Ind. Appl. (WARTIA)*, Sep. 2014, pp. 1175–1178.

[4] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining (WSDM)*, 2010, pp. 81–90.

[5] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 811–820.

[6] D. Li, C. Chen, T. Lu, S. Chu, and N. Gu, "Mixture matrix approximation for collaborative filtering," *IEEE Trans. Knowl. Data Eng.*, to be published.

[7] D. Li, C. Chen, Q. Lv, H. Gu, T. Lu, L. Shang, N. Gu, and S. M. Chu, "AdaError: An adaptive learning rate method for matrix approximation-based collaborative filtering," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 741–751.

[8] H. Wu and G. Liu, "A hybrid model on learning cross features for transaction fraud detection," in *Proc. 19th Ind. Conf. Data Mining*, 2019, pp. 88–102.

[9] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 995–1000.

[10] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for CTR prediction," in *Proc. 10th ACM Conf. Recommender Syst. (RecSys)*, 2016, pp. 43–50.

[11] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1725–1731.

[12] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3119–3125.

[13] J. Sun, A. Vishnu, A. Chakrabarti, C. Siegel, and S. Parthasarathy, "ColdRoute: Effective routing of cold questions in stack exchange sites," *Data Mining Knowl. Discovery*, vol. 32, no. 5, pp. 1339–1367, Jun. 2018.

[14] F. M. Harper, D. Raban, S. Rafaeli, and J. A. Konstan, "Predictors of answer quality in online Q&A sites," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2008, pp. 865–874.

[15] G. Wang, K. Gill, M. Mohanlal, H. Zheng, and B. Y. Zhao, "Wisdom in the social crowd: An analysis of Quora," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, 2013, pp. 1341–1352.

[16] G. Li, H. Zhu, T. Lu, X. Ding, and N. Gu, "Is it good to be like Wikipedia?: Exploring the trade-offs of introducing collaborative editing model to Q&A sites," in *Proc. 18th ACM Conf. Comput. Supported Cooperat. Work Social Comput.*, 2015, pp. 1080–1091.

[17] G. Li, T. Lu, X. Ding, and N. Gu, "Predicting collaborative edits of questions and answers in online Q&A sites," *J. Internet Technol.*, vol. 17, no. 6, pp. 1187–1194, 2016.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, and A. N. Gomez, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[19] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: Structure and algorithms," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, 2007, pp. 221–230.

[20] J. Yang, L. A. Adamic, and M. S. Ackerman, "Competing to share expertise: The Taskcn knowledge sharing community," in *ICWSM*, 2008, pp. 1–8.

[21] J. Liu, Y.-I. Song, and C.-Y. Lin, "Competition-based user expertise score estimation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. (SIGIR)*, 2011, pp. 425–434.

[22] Z. Zhao, F. Wei, M. Zhou, and W. Ng, "Cold-start expert finding in community question answering via graph regularization," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2015, pp. 21–38.

[23] H. Zhu, E. Chen, H. Xiong, H. Cao, and J. Tian, "Ranking user authority with relevant knowledge categories for expert finding," *World Wide Web*, vol. 17, no. 5, pp. 1081–1107, Apr. 2013.

[24] Z. Liu, K. Li, and D. Qu, "Knowledge graph based question routing for community question answering," in *Proc. Int. Conf. Neural Inf. Process.*, 2017, pp. 721–730.

[25] J. Sun, S. Moosavi, R. Ramnath, and S. Parthasarathy, "QDEE: Question difficulty and expertise estimation in community question answering sites," in *Proc. 12th Int. AAAI Conf. Web Social Media*, 2018, pp. 375–384.

[26] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen, "CQArank: Jointly model topics and expertise in community question answering," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 99–108.

[27] T. Zhao, N. Bian, C. Li, and M. Li, "Topic-level expert modeling in community question answering," in *Proc. SIAM Int. Conf. Data Mining*, Dec. 2013, pp. 776–784.

[28] S. Mumtaz, C. Rodriguez, and B. Benatallah, "Expert2Vec: Experts representation in community question answering for question routing," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2019, pp. 213–229.

[29] M. Bouguessa, B. Dumoulin, and S. Wang, "Identifying authoritative actors in question-answering forums: The case of yahoo! Answers," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 866–874.

[30] M. Bouguessa and L. B. Romdhane, "Identifying authorities in online communities," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–23, Apr. 2015.

[31] T. P. Sahu, N. K. Nagwani, and S. Verma, "Multivariate beta mixture model for automatic identification of topical authoritative users in community question answering sites," *IEEE Access*, vol. 4, pp. 5343–5355, 2016.

[32] B. V. Hanrahan, G. Convertino, and L. Nelson, "Modeling problem difficulty and expertise in stackoverflow," in *Proc. ACM Conf. Comput. Supported Cooperat. Work Companion (CSCW)*, 2012, pp. 91–94.

[33] A. Huna, I. Srba, and M. Bielikova, "Exploiting content quality and question difficulty in CQA reputation systems," in *Proc. Int. Conf. School Netw. Sci.*, 2016, pp. 68–81.

[34] J. Yang, K. Tao, A. Bozzon, and G.-J. Houben, "Sparrows and owls: Characterisation of expert behaviour in StackOverflow," in *Proc. Int. Conf. Modeling, Adaptation, Personalization*, 2014, pp. 266–277.

[35] 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. IJCAI*, Jul. 2016, pp. 3000–3006.

[36] X. Cheng, S. Zhu, S. Su, and G. Chen, "A multi-objective optimization approach for question routing in community question answering services," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1779–1792, Sep. 2017.

[37] T. Chen, H. Yin, H. Chen, R. Yan, Q. V. H. Nguyen, and X. Li, "AIR: Attentional intention-aware recommender systems," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 304–315.

[38] Q. Zhu, X. Zhou, Z. Song, J. Tan, and L. Guo, "DAN: Deep attention neural network for news recommendation," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 5973–5980.

[39] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, and M. Wang, "A hierarchical attention model for social contextual image recommendation," *IEEE Trans. Knowl. Data Eng.*, to be published.

[40] C. Zhou, J. Bai, J. Song, X. Liu, Z. Zhao, and X. Chen, "ATRank: An attention-based user behavior modeling framework for recommendation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4564–4571.

[41] C. Wu, F. Wu, S. Ge, T. Qi, Y. Huang, and X. Xie, "Neural news recommendation with multi-head self-attention," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 6390–6395.

[42] D. Cong, Y. Zhao, B. Qin, Y. Han, M. Zhang, A. Liu, and N. Chen, "Hierarchical attention based neural network for explainable recommendation," in *Proc. the Int. Conf. Multimedia Retr. - ICMR*, 2019, pp. 373–381.

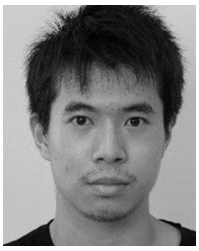
[43] H. Wang, G. Liu, A. Liu, Z. Li, and K. Zheng, "DMRAN: A hierarchical fine-grained attention-based network for recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3698–3704.

[44] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval* vol. 463. New York, NY, USA: ACM, 1999.

[45] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[46] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 404–411.

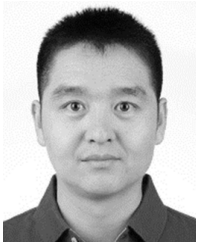
[47] Z. Li, J.-Y. Jiang, Y. Sun, and W. Wang, "Personalized question routing via heterogeneous network embedding," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, pp. 192–199.



WEIZHAO TANG received the B.S. degree in physics from Fudan University, Shanghai, China, in 2012, where he is currently pursuing the Ph.D. degree in computer science with the School of Computer Science. His current research interests include machine learning, data mining techniques, and recommender systems.

TUN LU received the Ph.D. degree in computer science from Sichuan University, in 2006.

He was a Visiting Scholar with the HCI Institute, Carnegie Mellon University, USA, in 2015. He is currently an Associate Professor with the School of Computer Science, Fudan University, Shanghai, China. He has published more than 60 peer-reviewed publications in prestigious journals and conferences such as CSCW, CHI, WWW, NIPS, UbiComp, and so on. His research interests include computer supported cooperative works (CSCW), social computing, and human–computer interaction (HCI). He shared a Best Paper Award at CSCW'15 and an Honorable Mention Award at CSCW'18. He is a member of ACM and a Senior Member of China Computer Federation (CCF). He is the Secretary General of CCF Technical Committee of Cooperative Computing. He has been active in professional services by serving as the PC Co-Chair (e.g., ChineseCSCW'17 & 18 & 19 and CSCWD'10), an Associate Chair (e.g., CHI'19 & 20 and CSCW'19 & 20), a PC Member (e.g., GROUP'18, CRIWG'17 & 2018, and CSCWD'16), a Guest Editor (e.g., *International Journal of Cooperative Information Systems* and the *Chinese Journal of Computers*) and reviewers for many well-known journals and conferences.



DONGSHENG LI (Member, IEEE) received the Ph.D. degree from the School of Computer Science, Fudan University, Shanghai, China, in 2012. He has been a Research Staff Member with IBM Research–China, since April 2015. He is currently an Adjunct Professor with the School of Computer Science, Fudan University. His research interests include recommender systems and general machine learning applications. In April 2018, he received one of the highest technical awards in IBM – the IBM Corporate Award.

HANSU GU received the B.S. degree in computer science from Fudan University, Shanghai, China, in 2008, and the Ph.D. degree in electrical engineering from the University of Colorado Boulder, Boulder, CO, USA, in 2013. He is currently a Machine Learning Scientist with Microsoft Inc., Seattle, USA, working on targeted advertising using deep learning. His research interests include text mining and recommendation systems.

NING GU received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 1995.

He is currently a Professor and the Director of the Cooperative Information and Systems Laboratory, School of Computer Science, Fudan University, Shanghai, China. His research interests include human-centered cooperative computing, CSCW and social computing, and human–computer interaction.